Contents

Plan Your Deployment ........................................................................... 1
Get Started with Tableau Server on Linux ........................................... 1
Server Administrator Overview ............................................................ 3
Architectural overview ........................................................................ 3
Tableau and your data .......................................................................... 5
User access .......................................................................................... 6
Where should I install Tableau Server in my network? ......................... 6
Sizing and scalability ........................................................................... 7
Tableau Server management model ....................................................... 8
Management tools ............................................................................... 8
Security ............................................................................................... 9
Tableau Services Manager Overview .................................................... 9
Functionality ....................................................................................... 10
Components ......................................................................................... 10
TSM Authentication ............................................................................. 11
Custom PAM service definition .......................................................... 11
Connecting TSM clients ....................................................................... 12
Infrastructure Planning ......................................................................... 13
Before you install... ............................................................................ 13
Hardware recommendations ................................................................. 13
Two Node Installation - Specialized for extract heavy environments .......... 29
Recommendations ............................................................................. 29
Server Configuration ........................................................................ 31
Two Node Installation - Specialized for flow environments .................. 34
Server Configuration ........................................................................ 35
High Availability Installations (HA) ...................................................... 38
Recommendations ............................................................................. 38
Server Configuration ........................................................................ 38
Virtual Machines and Public Cloud Deployments ................................. 41
Beyond Baseline Configurations ......................................................... 41
Disaster Recovery Considerations ...................................................... 41
Tableau Server Scalability .................................................................. 42
Identity Store ..................................................................................... 43
Local identity store ........................................................................... 43
External identity store ....................................................................... 43
LDAP bind ......................................................................................... 44
LDAP with GSSAPI (Kerberos) bind ................................................... 45
LDAP over SSL ................................................................................ 45
System user and groups .................................................................... 46
User ................................................................................................. 47
Groups ............................................................................................ 47
Authenticating clients ........................................................................ 48
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Configure Initial Node Settings</td>
<td>106</td>
</tr>
<tr>
<td>Prerequisite</td>
<td>106</td>
</tr>
<tr>
<td>Use the TSM web interface</td>
<td>106</td>
</tr>
<tr>
<td>Identity store settings</td>
<td>108</td>
</tr>
<tr>
<td>Gateway port</td>
<td>111</td>
</tr>
<tr>
<td>Product usage data</td>
<td>111</td>
</tr>
<tr>
<td>Sample workbook installation</td>
<td>112</td>
</tr>
<tr>
<td>Initializing install</td>
<td>112</td>
</tr>
<tr>
<td>Use the TSM CLI</td>
<td>113</td>
</tr>
<tr>
<td>Configure identity store settings</td>
<td>113</td>
</tr>
<tr>
<td>Configure gateway settings (optional)</td>
<td>114</td>
</tr>
<tr>
<td>Configure product usage data (optional)</td>
<td>114</td>
</tr>
<tr>
<td>Configure sample workbook installation (optional)</td>
<td>114</td>
</tr>
<tr>
<td>Apply pending configuration changes</td>
<td>115</td>
</tr>
<tr>
<td>Verify LDAP configuration (Optional)</td>
<td>115</td>
</tr>
<tr>
<td>Initialize and start Tableau Server</td>
<td>116</td>
</tr>
<tr>
<td>Next Step</td>
<td>116</td>
</tr>
<tr>
<td>Configuration File Example</td>
<td>117</td>
</tr>
<tr>
<td>Entities vs keys</td>
<td>118</td>
</tr>
<tr>
<td>Server Usage Data</td>
<td>119</td>
</tr>
<tr>
<td>Disable Usage Data Sharing</td>
<td>119</td>
</tr>
<tr>
<td>Disabling the sharing of usage data at install</td>
<td>119</td>
</tr>
</tbody>
</table>
Before you begin ........................................................................................................ 132

Step 1: Configure initial node. .......................................................... 132

Step 2: Configure additional nodes ......................................................... 133

Automated Installation of Tableau Server ............................................. 135

Benefits of using the automated installer ............................................ 135

Before you begin ....................................................................................... 136

How to use the automated installer .................................................... 137

Configure Tableau Server for a forward proxy .................................... 138

Perform an unattended installation .......................................................... 139

Step 1: Install the automated installer .................................................. 139

Step 2: Create files to provide additional configuration information required to run the automated install ................................................. 139

Step 3: Run the automated install .......................................................... 140

Install Tableau Server in a Disconnected (Air-Gapped) Environment ........ 142

Prerequisites .............................................................................................. 142

Installing Tableau Server on an Air-Gapped Computer Running Linux ........ 143

Activating Tableau Server in an Air-Gapped Environment .................. 146

Step 1. Transcribe data from your air-gapped system into an activation request template. ................................................................. 146

Step 2. Initialize or activate your license ................................................ 148

First time license activation ................................................................. 148

Subsequent license activation ............................................................... 148
Displaying Maps in an Air-Gapped Environment ........................................... 149

Install Tableau Server in the Cloud ................................................................. 150

Post Installation Tasks .................................................................................... 151

Security Hardening Checklist .......................................................................... 151

1. Update to the current version ................................................................. 151
2. Configure SSL/TLS with a valid, trusted certificate .............................. 151
3. Disable older versions of TLS ............................................................... 151
4. Configure SSL encryption for internal traffic ...................................... 152
5. Enable firewall protection ..................................................................... 153
6. Restrict access to the server computer and to important directories ...... 153
7. Generate fresh secrets and tokens ......................................................... 153
8. Disable services that you’re not using .................................................. 154
   REST API ....................................................................................... 154
   JMX Service .................................................................................. 155
9. Verify session lifetime configuration .................................................... 155
10. Configure a server safelist for file-based data sources ...................... 156
11. Enable HTTP Strict Transport Security for web browser clients .......... 156
12. Disable Guest access .......................................................................... 157
13. Set referrer-policy HTTP header to 'same-origin' .............................. 158
    Change List .................................................................................... 159
Configure SMTP Setup .................................................................................. 159
Secure SMTP ........................................................................................... 159
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Use the TSM web interface</td>
<td>160</td>
</tr>
<tr>
<td>Use the TSM CLI</td>
<td>161</td>
</tr>
<tr>
<td>Configuration file reference</td>
<td>162</td>
</tr>
<tr>
<td>TLS ciphers</td>
<td>166</td>
</tr>
<tr>
<td>Files and Permissions in TSM</td>
<td>168</td>
</tr>
<tr>
<td>Setting permissions for individual files</td>
<td>170</td>
</tr>
<tr>
<td>Setting permissions for directories</td>
<td>170</td>
</tr>
<tr>
<td>Configure Server Event Notification</td>
<td>171</td>
</tr>
<tr>
<td>Use the TSM web interface</td>
<td>171</td>
</tr>
<tr>
<td>Use the TSM CLI</td>
<td>174</td>
</tr>
<tr>
<td>Set notification values individually</td>
<td>174</td>
</tr>
<tr>
<td>Set all notification values with a single json file</td>
<td>176</td>
</tr>
<tr>
<td>Configure Data Cache</td>
<td>177</td>
</tr>
<tr>
<td>Database Drivers</td>
<td>178</td>
</tr>
<tr>
<td>Install drivers in a cluster</td>
<td>179</td>
</tr>
<tr>
<td>Server Crash Reporter</td>
<td>179</td>
</tr>
<tr>
<td>Configure Server Crash Reporter</td>
<td>180</td>
</tr>
<tr>
<td>Use the TSM web interface</td>
<td>180</td>
</tr>
<tr>
<td>Use the TSM CLI</td>
<td>181</td>
</tr>
<tr>
<td>Crash reporter settings</td>
<td>182</td>
</tr>
<tr>
<td>Configuration template</td>
<td>182</td>
</tr>
<tr>
<td>Configuration file reference</td>
<td>182</td>
</tr>
</tbody>
</table>
Distributed Installation Recommendations .......................................... 197

Recommendations for all installations .................................................... 197

Install and Configure Additional Nodes .................................................. 199

Use the TSM web interface .................................................................... 199

Generate the node bootstrap file ................................................................. 199

Install and initialize an additional node .................................................. 201

Configure the additional node with Cluster Controller .......................... 203

Use the TSM CLI .................................................................................. 205

Generate the node bootstrap file ................................................................. 205

Install and initialize an additional node .................................................. 206

Configure the additional node ................................................................. 209

Install drivers ....................................................................................... 210

Drivers and administrative views ......................................................... 211

Database Drivers .................................................................................. 211

Install drivers in a cluster ..................................................................... 211

Example: Install and Configure a Three-Node HA Cluster ...................... 212

A Single Server System ......................................................................... 212

A Three-Node System .......................................................................... 214

Configuration steps .............................................................................. 215

Before you begin .................................................................................... 216

Use the TSM web interface ................................................................... 216

Step 1: Install the initial node ................................................................. 216
Step 2: Generate a bootstrap file for the additional nodes ...................... 216
Step 3: Install and initialize an additional node .................................. 217
Step 4: Install and initialize a second additional node .......................... 220
Step 5: Configure the new node with a minimum topology .................... 223
Step 6: Deploy a Coordination Service ensemble ................................. 224
Step 7: Configure processes for the second node ................................ 228
Step 8: Configure processes for third node ....................................... 229

Use the TSM CLI ........................................................................... 230

Step 1: Install the initial node ......................................................... 230
Step 2: Generate a bootstrap file for the additional nodes .................. 230
Step 3: Install an additional node ................................................... 231
Step 4: Install a second additional node .......................................... 234
Step 5: Add a process to the additional nodes .................................... 237
Step 6: Deploy a Coordination Service ensemble ............................... 237
Step 7: Configure processes for the second node .............................. 242
Step 8: Configure processes for third node ..................................... 243
Step 9: Configure firewall rules (optional) ...................................... 244

Add a Load Balancer .................................................................. 244
Add a Load Balancer .................................................................. 244

Configure Tableau Server to work with a load balancer ................... 245
Load balancer configuration key-value pairs ................................. 245
Configuration template example .................................................. 247
Deploy a Coordination Service Ensemble ........................................... 248
Hardware requirements ............................................................... 248
The Coordination Service Quorum ............................................... 249
  Number of Coordination Service instances to use ....................... 249
Deploy a new Coordination Service ensemble ................................ 250
Configure Client File Service ...................................................... 254
  Configure CFS on additional nodes ........................................... 255
Repository Failover ..................................................................... 255
  Automatic repository failover ................................................... 256
  Manual repository failover ....................................................... 256
  Preferred active repository ...................................................... 257
Recover from an Initial Node Failure .............................................. 257
  General requirements ............................................................. 258
  Move the TSM Controller and License Service to another node .... 259
Recover from a Node Failure ......................................................... 263
  General requirements ............................................................. 264
  Removing a Failed Node .......................................................... 264
Configure Nodes ......................................................................... 267
  Use the TSM web interface ....................................................... 267
  Use the TSM CLI ................................................................... 269
    Adding processes to a node ...................................................... 270
    Changing the number of processes on a node ......................... 271
Adding a second instance of File Store ........................................ 285
Decommissioning and removing an instance of file store .............. 286
Use the TSM CLI ........................................................................ 287
Adding a second instance of file store ........................................ 287
Decommissioning and removing an instance of File Store .......... 287
Move the Messaging Service Process .......................................... 288
Use the TSM web interface ....................................................... 288
Moving the Messaging Service .................................................... 289
Use the TSM CLI ........................................................................ 289
Moving the Messaging Service .................................................... 289
Remove a Node .......................................................................... 290
Prerequisites for removing a node ............................................. 290
Use the TSM web interface ....................................................... 291
Use the TSM CLI ........................................................................ 292
Configure Tableau Server for High Availability with Coordination Service-Only Nodes .............................................. 293
Prerequisite ............................................................................... 293
Deploy an ensemble on Coordination Service-only nodes ............ 294
Add a Load Balancer ................................................................. 298
Configure Tableau Server to work with a load balancer ............... 299
Load balancer configuration key-value pairs ............................... 299
Configuration template example ............................................... 301
Tableau Server on Linux Administrator Guide

Post export steps ................................................................. 316

Step 1: Copy the zip file into the Tableau Server directory .................. 316

Step 2: Generate and validate the mapping files ............................ 317

Step 3: Import the correctly mapped files to the target site .............. 317

Troubleshooting ............................................................... 318

If you continue to publish new content to your source site after the export .. 318

If you encounter difficulty importing your site into Tableau Server ........ 318

Migrate Tableau Server from Windows to Linux .......................... 318

Step 1: Plan your migration ................................................. 319

Step 2: Create a backup .................................................... 321

  Migrate from 2018.1 or earlier ........................................... 321

  Migrate from 2018.2 or later ............................................. 321

Step 3: Install Tableau Server on Linux and restore the Windows backup ... 322

Step 4: Test Tableau Server on Linux .................................... 323

Step 5: Install Tableau Server on Linux in your production environment and restore the Windows backup ........................................... 325

Test Workbook Performance .................................................. 325

  Set up your test environment .............................................. 326

  Capture performance metrics and analyze ................................ 326

  Troubleshoot performance issues ....................................... 327

  Capture the conclusions and results from your testing ............... 330

Migrate from Tabadmin to the TSM CLI .................................. 330
Tabadmin commands with a corresponding TSM CLI command ............ 330
Tabadmin commands with no corresponding TSM CLI command .......... 335
Migrate Tableau Server from an On-Premises Computer to a VM in the Cloud .. 337
Prerequisites ..................................................................................... 337
To migrate Tableau Server to a VM in the cloud .................................. 338
Upgrade Tableau Server Overview ....................................................... 341
Upgrade Tableau Server on Linux ....................................................... 342
Research the Upgrade ........................................................................ 342
What's New in Tableau Server ......................................................... 345
What's new in version 2019.4 .............................................................. 345
Request access ................................................................................... 345
Configure recommendations for views .............................................. 346
Enable or disable Ask Data for a site ................................................ 346
Personal access tokens ....................................................................... 346
Secure SMTP ....................................................................................... 347
Test Email Configuration ................................................................... 347
Create Extracts on the Web ............................................................... 347
New Installers for Tableau Server and tabcmd ................................. 347
Custom welcome banner .................................................................... 347
Limit visibility of user information .................................................... 348
Manage Sandboxed Extensions .......................................................... 348
Added in previous versions ............................................................... 348
<table>
<thead>
<tr>
<th>Feature</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add PDF Attachments to Subscription Emails</td>
<td>349</td>
</tr>
<tr>
<td>Tableau Catalog features—part of the Data Management Add-on</td>
<td>349</td>
</tr>
<tr>
<td>Tableau Catalog on the server—part of the Data Management Add-on</td>
<td>350</td>
</tr>
<tr>
<td>Tableau Metadata API</td>
<td>350</td>
</tr>
<tr>
<td>Job Management</td>
<td>350</td>
</tr>
<tr>
<td>Support for Embedded Credentials in Bootstrap Files</td>
<td>350</td>
</tr>
<tr>
<td>Server Management Add-on</td>
<td>351</td>
</tr>
<tr>
<td>Support for Debian 9 Linux Distributions</td>
<td>352</td>
</tr>
<tr>
<td>Extract Encryption at Rest</td>
<td>352</td>
</tr>
<tr>
<td>Date-range Support for Ziplogs</td>
<td>352</td>
</tr>
<tr>
<td>Tableau Server Usage Data</td>
<td>353</td>
</tr>
<tr>
<td>Add a Note to Tableau Server Sign In Page</td>
<td>353</td>
</tr>
<tr>
<td>Support added for Italian</td>
<td>354</td>
</tr>
<tr>
<td>Direct access with TSM CLI</td>
<td>356</td>
</tr>
<tr>
<td>Local authentication password policy improvements</td>
<td>356</td>
</tr>
<tr>
<td>Select from ten different languages for Tableau</td>
<td>356</td>
</tr>
<tr>
<td>Set a site-level start page</td>
<td>358</td>
</tr>
<tr>
<td>Elastic Server on multiple nodes</td>
<td>358</td>
</tr>
<tr>
<td>SSL connection to TabPy external service</td>
<td>358</td>
</tr>
<tr>
<td>Configuring the threshold for suspending data-driven alerts</td>
<td>358</td>
</tr>
<tr>
<td>Web authoring auto recovery</td>
<td>359</td>
</tr>
<tr>
<td>Option to install to non-default location on RHEL-distros</td>
<td>360</td>
</tr>
<tr>
<td>Feature</td>
<td>Page</td>
</tr>
<tr>
<td>------------------------------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>Microservice Container Processes</td>
<td>360</td>
</tr>
<tr>
<td>Pluggable Authentication Modules (PAM)</td>
<td>360</td>
</tr>
<tr>
<td>TSM client authentication</td>
<td>360</td>
</tr>
<tr>
<td>Ask Data and Elastic Server Processes</td>
<td>361</td>
</tr>
<tr>
<td>Tableau Prep Conductor Server Process</td>
<td>361</td>
</tr>
<tr>
<td>Tableau Data Management Add-On</td>
<td>361</td>
</tr>
<tr>
<td>Node roles for Backgrounder Process</td>
<td>361</td>
</tr>
<tr>
<td>SSL connection to Rserve external service</td>
<td>362</td>
</tr>
<tr>
<td>Materialize calculations in extracts already published to Tableau Server</td>
<td>363</td>
</tr>
<tr>
<td>Secure extract data at the row level</td>
<td>363</td>
</tr>
<tr>
<td>Improved content browsing (in-release beta)</td>
<td>363</td>
</tr>
<tr>
<td>Support for Content Security Policy</td>
<td>364</td>
</tr>
<tr>
<td>Improved support for LDAP</td>
<td>364</td>
</tr>
<tr>
<td>Connections to external services support SSL</td>
<td>364</td>
</tr>
<tr>
<td>SAML now supports PKCS#8 key files</td>
<td>364</td>
</tr>
<tr>
<td>Improved key and secrets refresh process</td>
<td>364</td>
</tr>
<tr>
<td>Backup without compression (added in version 2018.3.2)</td>
<td>365</td>
</tr>
<tr>
<td>Include colleagues on data conversations with @mentions</td>
<td>365</td>
</tr>
<tr>
<td>Create a support case from the TSM UI or CLI</td>
<td>366</td>
</tr>
<tr>
<td>Manage Dashboard Extensions in Tableau Server</td>
<td>366</td>
</tr>
<tr>
<td>Automated daemon port configuration</td>
<td>366</td>
</tr>
<tr>
<td>Amazon Linux 2 now supported</td>
<td>367</td>
</tr>
</tbody>
</table>
Resize headers on the web ................................................................. 367
Streamline discussion threads by deleting comments ..................... 368
New user-based term licenses are now available .......................... 368
New and renamed site roles are now available ............................... 368
Improved support for managing web data connectors ................... 368
Optimized server performance for data alerts ............................... 369
Nested projects ............................................................................... 370
Filtered search hidden by default ................................................. 370
Tableau Data Engine with Hyper ................................................. 371
Extracts use the new .hyper format ............................................. 371
Changes to the way values are computed ..................................... 371
Tableau Services Manager ............................................................ 372
Hot topology support ..................................................................... 372
Lightweight Directory Access Protocol (LDAP) support .................. 372
What's Changed - Things to Know Before You Upgrade .................. 373
New tabcmd Installer for Windows .............................................. 373
Updates to Tableau Server Job Management ................................ 374
Default cipher suite no longer includes default support for old ciphers ............................................. 375
Tableau Server Service Manager has been removed .................... 376
Usage data is sent to Tableau ....................................................... 376
Upgrade without needing to provide a password ......................... 377
Some tsm command options are no longer configurable ............... 377
Minimum hardware requirements have changed ........................................ 378
Upgrade requires at least 15% free disk space on each node ...................... 378
"Referrer-Policy" HTTP header added ................................................. 378
Map-related internet addresses have changed .................................... 379
Install to non-default locations (RHEL-like distributions only) .................. 379
High-resolution thumbnails .................................................................. 379
Project card images ........................................................................... 380
Floating worksheet captions become transparent .................................... 381
Changes to some Data Engine tsm configuration set options .................... 381
SSL Off-loading and SAML .................................................................. 381
Moving the repository in a two-node cluster not supported ...................... 382
Updates to Tableau Server Apache Logs .............................................. 382
The Read Only site role is now deprecated .......................................... 383
The tabcmd initialuser command prompts for password ....................... 383
New ports added ............................................................................... 383
  Automatic port assignment .............................................................. 383
  Manual port assignment ................................................................ 384
  Ports added in 2018.2 .................................................................... 384
Removed tabadmin command options .................................................. 385
Updated tsm commands ................................................................. 385
Updated upgrade-tsm script ............................................................... 385
Tableau Server configuration changes .................................................. 387
Changes to extracts ................................................................. 387
Version compatibility ............................................................. 388
Viewer (can publish) and Unlicensed (can publish) site roles removed .................................................. 388
Nested projects and content navigation .................................. 388
Nested projects and tabcmd ..................................................... 389
Filtered search hidden by default ............................................. 389
Minimum Hardware Requirements and Recommendations for Tableau Server ............................................. 390
Minimum Hardware Requirements for Testing and Prototyping ................................................................. 391
Minimum Hardware Recommendations .................................... 392
Prepare for the Upgrade .......................................................... 394
Gather custom configuration information ................................ 394
Gather the environment configuration ........................................ 396
Copy the setup files to a local file share .................................... 396
Update tabcmd ........................................................................ 397
Back up Tableau Server data .................................................... 397
Generate asset keys file before uninstalling Tableau Server for Windows 2018.1.x or earlier ............................................. 398
Locating an existing asset keys file ............................................ 398
Restoring a pre-2018.2 Tableau Server for Windows backup along with asset keys ............................................. 399
Check your product maintenance status .................................... 399
Disable your extract refreshes ................................................... 399
Plan your strategy if you use virtual machines (VMs) ....................... 400

Test the Upgrade ................................................................. 400

Prepare a test environment ...................................................... 401

Upgrade the test environment .................................................. 402

Confirm that everything works as expected ............................... 402

Performance and user acceptance testing ................................. 403

Test new features .................................................................. 403

Communicate about the upgrade .............................................. 404

Test Workbook Performance .................................................. 404

Set up your test environment .................................................. 405

Capture performance metrics and analyze ............................... 405

Troubleshoot performance issues .......................................... 406

Capture the conclusions and results from your testing ............... 408

Perform the Upgrade .............................................................. 409

Upgrade Tableau Server on Linux ............................................. 409

Upgrading Tableau Server on Linux ......................................... 412

Related topics ..................................................................... 415

Upgrade Tableau Server on Linux from 10.5 ............................ 416

Upgrade to 10.5.x ................................................................ 417

Install 2018.x or later ............................................................. 418

Run TSM commands .............................................................. 419

Migrate 10.5.x to single user ................................................. 420
Upgrade to 2018.x or later ................................................................. 420

Related topics ............................................................................. 422

Troubleshoot Tableau Server Install and Upgrade ....................... 422

General Troubleshooting Steps .................................................... 422

Common Tableau Server Install Issues ........................................ 423

Installation logs location ............................................................... 423

Unable to log into TSM or Tableau Server (sign in screen redisplay after entering credentials) ........................................ 423

Multiple install attempts fail ....................................................... 423

Install fails due to hardware requirements .................................. 424

Common Tableau Server Upgrade Issues .................................... 425

Upgrade logs location ................................................................. 425

Upgrade multi-node, initializing additional node fails with "Enter your credentials again" error ........................................... 425

Upgrading fails due to lack of disk space .................................... 425

Troubleshooting connections to TSM .......................................... 426

Starting Tableau Server .............................................................. 426

Tableau Server cannot determine if it fully started ..................... 426

Tableau Server doesn't start ....................................................... 426

Reindexing Tableau Server Search & Browse ............................ 427

Problems that can be solved by rebuilding Search & Browse index 427

Activating Tableau Server .......................................................... 427
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Importing at the server level in multi-site environments</td>
<td>471</td>
</tr>
<tr>
<td>Importing to a single-site environment</td>
<td>472</td>
</tr>
<tr>
<td>Multi-site versus single-site import</td>
<td>472</td>
</tr>
<tr>
<td>CSV Import File Guidelines</td>
<td>473</td>
</tr>
<tr>
<td>CSV file format requirements</td>
<td>473</td>
</tr>
<tr>
<td>Required columns in the CSV file</td>
<td>474</td>
</tr>
<tr>
<td>Additional import file options</td>
<td>474</td>
</tr>
<tr>
<td>Improve performance for large CSV files passed through tabcmd</td>
<td>475</td>
</tr>
<tr>
<td>Notes</td>
<td>476</td>
</tr>
<tr>
<td>CSV settings and site roles</td>
<td>476</td>
</tr>
<tr>
<td>CSV import example for Tableau Server</td>
<td>477</td>
</tr>
<tr>
<td>Manage Site User Visibility</td>
<td>478</td>
</tr>
<tr>
<td>Limit user visibility</td>
<td>478</td>
</tr>
<tr>
<td>Best practices for limiting user visibility</td>
<td>480</td>
</tr>
<tr>
<td>User visibility in the Tableau Metadata API</td>
<td>481</td>
</tr>
<tr>
<td>Restore Full User Visibility</td>
<td>481</td>
</tr>
<tr>
<td>Guest User</td>
<td>481</td>
</tr>
<tr>
<td>Guest user permissions</td>
<td>482</td>
</tr>
<tr>
<td>Enable or disable Guest access</td>
<td>482</td>
</tr>
<tr>
<td>Additional Guest account characteristics</td>
<td>483</td>
</tr>
<tr>
<td>Groups</td>
<td>484</td>
</tr>
<tr>
<td>Add Users to a Group</td>
<td>484</td>
</tr>
</tbody>
</table>
Add users to a group (Users page) ................................................. 485
Add users to a group (Groups page) ................................................. 485
Create a Local Group ................................................................... 486
Create Groups via Active Directory ............................................ 487
Synchronize Active Directory Groups in a Site ......................... 489
Set the minimum site role for users in an Active Directory group ... 490
What happens when users are removed in the source Active Directory? ... 491
What happens when a user name changes in the source Active Directory ... 491
What happens when an Active Directory group is removed from Tableau Server? ......................................................................... 492
Synchronize All Active Directory Groups on the Server ............... 492
Before you begin ........................................................................ 493
Synchronize Active Directory groups on a schedule .................... 493
Synchronize all Active Directory groups on demand .................. 494
View synchronization activity .......................................................... 495
Set the minimum site role for users in an Active Directory group ... 495
What happens when users are removed in the source Active Directory? ... 497
Improving group synchronization performance .......................... 497
Quick Start: Synchronize All Active Directory Groups on a Schedule .......... 497
1 Set a minimum site role for synchronization .............................. 498
2 Set the schedule ...................................................................... 499
3 Run synchronization on-demand (optional) ............................... 499
4 View the status of synchronization tasks ........................................... 500

Delete Groups .................................................................................... 501

Dashboard-based Custom Portals ......................................................... 502

Sketch out a portal design ................................................................. 503

Gather images for logos and navigation elements .............................. 503

Lay out text, images, and selected sheets on a dashboard .................. 504

Link dashboard elements to content ................................................ 504

Publish, test, and refine your portal .................................................. 505

Manage Content Access .................................................................... 506

Set a Site’s Web Authoring Access and Functions ............................. 506

Turn web authoring on or off for a site .............................................. 506

Notes .................................................................................................. 507

See which sites allow web authoring ................................................ 507

Configure cross-database join options .............................................. 508

Set Web Edit, Save, and Download Access on Content ...................... 509

Why allow users to work on the site directly ...................................... 509

Web authoring pros and cons ............................................................ 509

Managing permissions to help users avoid content proliferation ....... 510

Coordinate edit and save capabilities with site roles for the appropriate level of access ............................................................................................................. 510

Site role access .................................................................................. 511

Configure Projects, Groups, and Permissions for Managed Self-Service .... 511
Create a project team and adopt a permissions strategy .................................. 512
Steps to coordinate projects and groups ............................................................ 512
1. Plan your permissions .................................................................................... 513
   Use a closed permissions model for managed content .................................. 514
2. Remove permissions that will cause ambiguities ............................................ 514
3. Create groups ................................................................................................ 516
4. Assign permissions to the groups .................................................................. 517
   Set permissions templates ............................................................................. 517
   Example ......................................................................................................... 518
5. Create projects and adjust permissions .......................................................... 519
   Example project structure ............................................................................. 519
   Help project leaders manage content and users find it ................................. 521
6. Lock permissions in each project ................................................................. 522
   Automate working with groups and projects ............................................... 522
   Next steps ..................................................................................................... 523
   Learn how Tableau and some of our customers address governance and
   self-service .................................................................................................. 524
Use Projects to Manage Content Access ............................................................ 524
   Why use projects .......................................................................................... 525
   When to create project hierarchies (example) ............................................... 526
   Why not use sites? ......................................................................................... 527
   Project-level administration ............................................................................ 527
Data Roles ................................................................. 546
Set permissions ....................................................... 546
Set Project Permissions ........................................... 547
Set Content Permissions .......................................... 548
Set permissions on content ..................................... 548
Set permissions on a view ....................................... 550
Set Permissions on Publish ...................................... 550
Permission capabilities .......................................... 552
Permission Roles ..................................................... 552
Projects .................................................................. 552
  Capabilities: .......................................................... 553
  Permission roles: .................................................. 554
Workbooks ............................................................... 554
  Capabilities: .......................................................... 556
  Permission roles: .................................................. 557
Data Sources ........................................................... 557
  Capabilities: .......................................................... 558
  Permission roles: .................................................. 559
Flows and Data Roles ............................................. 560
  Capabilities: .......................................................... 561
  Permission roles: .................................................. 562
Data Roles ............................................................... 562
<table>
<thead>
<tr>
<th>Topic</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Performing tasks on Tableau Server 10.5</td>
<td>603</td>
</tr>
<tr>
<td>Downloading from Tableau Server 10.5</td>
<td>604</td>
</tr>
<tr>
<td>Automating refresh and append tasks</td>
<td>605</td>
</tr>
<tr>
<td>Why keep an extract in .tde format?</td>
<td>606</td>
</tr>
<tr>
<td>How to keep an extract in .tde format</td>
<td>607</td>
</tr>
<tr>
<td>What to expect after extract upgrade</td>
<td>607</td>
</tr>
<tr>
<td>Create Extracts on the Web</td>
<td>607</td>
</tr>
<tr>
<td>Limitations</td>
<td>607</td>
</tr>
<tr>
<td>Extract a Published Data Source</td>
<td>608</td>
</tr>
<tr>
<td>Extract a Data Source that is Embedded in a Published Workbook</td>
<td>608</td>
</tr>
<tr>
<td>View Data Source Attributes</td>
<td>609</td>
</tr>
<tr>
<td>View data sources by name</td>
<td>609</td>
</tr>
<tr>
<td>View a list of connections</td>
<td>610</td>
</tr>
<tr>
<td>Use Certification to Help Users Find Trusted Data</td>
<td>610</td>
</tr>
<tr>
<td>How certification helps users find trusted data</td>
<td>611</td>
</tr>
<tr>
<td>Create guidelines for selecting data to certify</td>
<td>612</td>
</tr>
<tr>
<td>Who can certify data</td>
<td>612</td>
</tr>
<tr>
<td>How to certify data</td>
<td>613</td>
</tr>
<tr>
<td>Certify data sources</td>
<td>613</td>
</tr>
<tr>
<td>Certify databases and tables</td>
<td>614</td>
</tr>
<tr>
<td>Keep Data Fresh</td>
<td>615</td>
</tr>
<tr>
<td>Manage Refresh Tasks</td>
<td>615</td>
</tr>
</tbody>
</table>
See also ..............................................................................................................616

Refresh Data on a Schedule ..............................................................................616

Quick Start: Refresh Extracts on a Schedule .................................................617

1 Set up a schedule on the server .................................................................617

2 Enable scheduled extract refreshes and failure emails ...............................618

3 Publish a workbook with an extract ..........................................................619

4 Monitor refresh performance .......................................................................620

Automate Refresh Tasks ..................................................................................621

Handle Extract Refresh Alerts ........................................................................621

Resolving Extract Refresh Problems ..............................................................623

Edit Connections on Tableau Server ..............................................................624

Authentication types for Google, Salesforce, and WDC data ......................626

Google authentication options ........................................................................627

Salesforce.com authentication options ...........................................................627

Monitor progress .............................................................................................628

Cube Data Sources ..........................................................................................629

Web Data Connectors in Tableau Server .......................................................629

Before you run connectors on Tableau Server .............................................630

Manage connectors in a safe list .................................................................630

Updating WDC safe lists requires a server restart ........................................631

Add connectors to the safe list and secondary safe list ..............................631

Allow or disallow WDCs or WDC extract refreshes ....................................632
Remove one or more WDCs from the safe list ........................................ 632

List all WDCs on the safe list ................................................................. 632

Refresh the extract for a connector .................................................... 632

Troubleshooting .................................................................................. 633

Testing and Vetting Web Data Connectors ........................................... 634

Examine the source ............................................................................ 634

Test the web data connector in an isolated environment .................... 635

Monitor the traffic created by the web data connector ....................... 635

Test the performance and resource usage of the web data connector ... 635

Enable Tableau Catalog ...................................................................... 635

Catalog on Tableau Online .................................................................. 636

Catalog on Tableau Server .................................................................. 636

Before enabling Catalog ..................................................................... 636

Step 1: Determine the amount of content on Tableau Server ............... 637

Step 2: Estimate how long ingestion will take ..................................... 637

Step 3 (optional): Decrease the time of ingestion and increase memory ... 638

Threads ............................................................................................... 638

Memory ............................................................................................... 638

Step 4: Activate the Data Management Add-on ................................. 639

Step 5 (optional): Turn off Catalog capabilities .................................... 639

Enabling Catalog ................................................................................ 639

Step 1: Run the tsm maintenance metadata-services enable command ... 639
<table>
<thead>
<tr>
<th>Step</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>Monitor ingestion progress</td>
</tr>
<tr>
<td>3</td>
<td>Configure SMTP Setup</td>
</tr>
<tr>
<td>4 (optional)</td>
<td>Turn on Catalog capabilities</td>
</tr>
<tr>
<td></td>
<td>Troubleshoot Catalog or the Metadata API</td>
</tr>
<tr>
<td></td>
<td>Timeout limit and node limit exceeded messages</td>
</tr>
<tr>
<td></td>
<td>Indexing performance</td>
</tr>
<tr>
<td></td>
<td>Disable Tableau Catalog</td>
</tr>
<tr>
<td></td>
<td>Stop indexing metadata on Tableau Server</td>
</tr>
<tr>
<td></td>
<td>Use Lineage for Impact Analysis</td>
</tr>
<tr>
<td></td>
<td>Navigate lineage</td>
</tr>
<tr>
<td></td>
<td>Lineage and custom SQL connections</td>
</tr>
<tr>
<td></td>
<td>Mismatch between lineage count and tab count</td>
</tr>
<tr>
<td></td>
<td>Workbook count mismatch example</td>
</tr>
<tr>
<td></td>
<td>Use email to contact owners</td>
</tr>
<tr>
<td></td>
<td>Set a Data Quality Warning</td>
</tr>
<tr>
<td></td>
<td>Who can see the data quality warning</td>
</tr>
<tr>
<td></td>
<td>Who can set a data quality warning</td>
</tr>
<tr>
<td></td>
<td>How to set a data quality warning</td>
</tr>
<tr>
<td></td>
<td>Remove a data quality warning</td>
</tr>
<tr>
<td></td>
<td>Manage Dashboard Extensions in Tableau Server</td>
</tr>
<tr>
<td></td>
<td>Before you run extensions on Tableau Server</td>
</tr>
<tr>
<td></td>
<td>Control dashboard extensions and access to data</td>
</tr>
</tbody>
</table>
Change the global setting enabling extensions on the server .................. 655
Change the default settings for a site .............................................. 655
Identifying an extension ............................................................... 655
Add extensions to the safe list and configure user prompts .................. 657
Block specific extensions .............................................................. 658
Test Network-enabled extensions for security ................................. 658
Examine the source files ............................................................. 658
Understand data access ............................................................... 659
Test the extension in an isolated environment .................................. 660
Monitor traffic created by the dashboard extension ............................ 660
Creators: Connect to data on the web ............................................. 660
Open the Connect to Data page ...................................................... 661
Tableau Server ............................................................................. 662
Connect to data On this site ......................................................... 662
Connect to files ........................................................................... 662
Use connectors ............................................................................ 662
Tableau Server connectors ............................................................ 663
Tableau Catalog Supported Connectors .......................................... 663
Tableau Online ............................................................................ 664
Connect to data On this site ......................................................... 664
Connect to files ........................................................................... 664
Use connectors ............................................................................ 664
Step 3: Configure AD FS to accept sign-in requests from Tableau Server 749

Step 4: Provide AD FS metadata to Tableau Server .......................... 753

Use SAML SSO with Kerberos Database Delegation .......................... 754

Overview of the process ................................................................. 754

Configure Tableau Server for SAML with Kerberos ....................... 755

Configure Site-Specific SAML ....................................................... 755

Prerequisites for enabling site-specific SAML ................................ 756

Server-wide settings related to site-specific SAML ......................... 757

Configure the server to support site-specific SAML ....................... 757

About the commands .................................................................... 758

Configure SAML for a site ............................................................ 758

Troubleshoot SAML ..................................................................... 763

SAML and Enable Automatic Logon ............................................. 763

HTTP Status 500 error when configuring SAML ......................... 763

Signing In from the Command Line ............................................. 764

Login fails: failed to find the user ................................................ 764

Login fails: SSL offloading ........................................................... 765

SAML Error Log ......................................................................... 765

Trailing Slash ............................................................................. 766

Confirm Connectivity ................................................................. 766

Kerberos .................................................................................... 766

How Kerberos works ................................................................... 767
Note 4: Mobile Safari or Tableau Mobile on Mac iOS ................................. 786
Note 5: Android platform ................................................................. 786

Troubleshoot Kerberos ................................................................. 786
Single sign-on to Tableau Server .................................................. 786

Troubleshooting sign-in errors on the client computer ....................... 787
Troubleshooting sign-in errors on the server ..................................... 789
Verify Kerberos configuration script ............................................... 790
Data source SSO ............................................................................ 790

Delegated data source access failures ............................................. 790
Kerberos delegation multi-domain configuration .................................. 791
Cross-domain constrained delegation ............................................. 792

Configure Mutual SSL Authentication ............................................. 792
Use the TSM web interface ............................................................. 793
Use the TSM CLI ............................................................................ 794
Step 1: Require SSL for external server communication ..................... 794
Step 2: Use mutual SSL ................................................................. 795
Additional options for mutual SSL .................................................. 796

Fallback authentication .................................................................... 796
User name mapping ........................................................................ 796

Certificate Revocation List (CRL) ................................................... 797
Mapping a Client Certificate to a User During Mutual Authentication .... 797
User-name mapping options ............................................................ 797
Step 1: Add a test user .......................................................... 830
Step 2: Create a test HTML page ........................................... 830
Step 3: Retrieve a trusted ticket from Tableau Server ............... 832
Step 4: Test access with trusted ticket .................................... 833
Troubleshoot Trusted Authentication .................................... 834
Ticket Value of -1 Returned from Tableau Server .................... 834
HTTP 401 - Not Authorized ................................................... 836
HTTP 404 - File Not Found ................................................... 836
Invalid User (SharePoint or C#) ........................................... 836
Attempting to Retrieve the Ticket from the Wrong IP Address .... 837
Cookie Restriction Error ..................................................... 837
An error occurred communicating with the server (403) .......... 837
Personal Access Tokens .................................................... 838
Understanding personal access tokens .................................. 839
Creating tokens ............................................................... 839
Revoking users' tokens ..................................................... 839
Tracking and monitoring usage .......................................... 840
Data Connection Authentication .......................................... 840
More information ............................................................ 840
Enable Kerberos Delegation ............................................... 841
Supported data sources .................................................... 841
Requirements ............................................................... 841
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Security</td>
<td>875</td>
</tr>
<tr>
<td>Best Practices for Row Level Security</td>
<td>877</td>
</tr>
<tr>
<td>Entitlement tables</td>
<td>878</td>
</tr>
<tr>
<td>Entitlement table models</td>
<td>878</td>
</tr>
<tr>
<td>Users and roles</td>
<td>879</td>
</tr>
<tr>
<td>Joins</td>
<td>879</td>
</tr>
<tr>
<td>Implement Row Level Security</td>
<td>880</td>
</tr>
<tr>
<td>Deepest granularity</td>
<td>880</td>
</tr>
<tr>
<td>Sparse entitlements</td>
<td>880</td>
</tr>
<tr>
<td>Data source filter</td>
<td>881</td>
</tr>
<tr>
<td>All access with deepest granularity</td>
<td>882</td>
</tr>
<tr>
<td>Performance and processing order of operations</td>
<td>883</td>
</tr>
<tr>
<td>Live connections</td>
<td>883</td>
</tr>
<tr>
<td>Extracts</td>
<td>884</td>
</tr>
<tr>
<td>Considerations with extracts</td>
<td>884</td>
</tr>
<tr>
<td>Single table extracts</td>
<td>885</td>
</tr>
<tr>
<td>Use built-in Row Level Security in a data source</td>
<td>886</td>
</tr>
<tr>
<td>Row Level Security in the Data Source</td>
<td>886</td>
</tr>
<tr>
<td>Impersonation (Microsoft SQL Server)</td>
<td>886</td>
</tr>
<tr>
<td>Kerberos &amp; constrained delegation</td>
<td>887</td>
</tr>
<tr>
<td>OLAP Cubes</td>
<td>887</td>
</tr>
<tr>
<td>SAML delegation &amp; SAP HANA</td>
<td>888</td>
</tr>
<tr>
<td>Section</td>
<td>Page</td>
</tr>
<tr>
<td>------------------------------------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>Initial SQL to force a user-specific session (Oracle VPD)</td>
<td>888</td>
</tr>
<tr>
<td>Comparison matrix for Row Level Security Methods</td>
<td>889</td>
</tr>
<tr>
<td>Manage Server Secrets</td>
<td>891</td>
</tr>
<tr>
<td>Understanding how secrets storage works</td>
<td>891</td>
</tr>
<tr>
<td>Who has access to the master key?</td>
<td>892</td>
</tr>
<tr>
<td>Import and export configuration information</td>
<td>893</td>
</tr>
<tr>
<td>Securing secrets for import and export operations</td>
<td>893</td>
</tr>
<tr>
<td>Cluster nodes</td>
<td>895</td>
</tr>
<tr>
<td>Secrets storage event logging</td>
<td>896</td>
</tr>
<tr>
<td>Managing secrets</td>
<td>896</td>
</tr>
<tr>
<td>Updating secrets</td>
<td>896</td>
</tr>
<tr>
<td>Retrieving passwords</td>
<td>897</td>
</tr>
<tr>
<td>Extension Security - Best Practices for Deployment</td>
<td>901</td>
</tr>
<tr>
<td>Security for extensions in Tableau</td>
<td>901</td>
</tr>
<tr>
<td>Sandboxed and Network-enabled extensions</td>
<td>901</td>
</tr>
<tr>
<td>Potential security risks with Network-enabled extensions</td>
<td>902</td>
</tr>
<tr>
<td>Mitigating the security threats with Network-enabled extensions</td>
<td>903</td>
</tr>
<tr>
<td>Manage extensions using Tableau</td>
<td>903</td>
</tr>
<tr>
<td>Recommendations for Tableau Desktop</td>
<td>904</td>
</tr>
<tr>
<td>Selectively enable extensions on Tableau Desktop</td>
<td>905</td>
</tr>
<tr>
<td>Recommendations for Tableau Server and Tableau Online</td>
<td>906</td>
</tr>
</tbody>
</table>
Trust Sandboxed extensions and the Network-enabled extensions on the safe list ................................................................. 906

Checklist for the safe list: ............................................................................. 907

Add extensions to the safe list: ................................................................. 907

  Block specific extensions from running on Tableau Server .................. 907

  Turn off extensions for a site ................................................................. 908

  Show or hide user prompts to run Network-enabled extensions ........... 908

  Turn off Sandboxed extensions ............................................................ 908

Extract Encryption at Rest ........................................................................ 909

Limitations ............................................................................................... 909

Performance Overview ............................................................................ 909

  Increase in Backgrounder Load .......................................................... 909

  Increase in Viz Load Time and Worker Load ...................................... 910

  Impact on Backup and Restore ............................................................ 911

Enforce Encryption at Rest on a Site ...................................................... 911

Enable Encryption at Rest on a Site ....................................................... 911

Disable Encryption at Rest on a Site ..................................................... 912

View Extract Encryption Mode for All Sites ............................................ 912

Encrypt or Decrypt Extracts for a Published Workbook or Data Source ... 912

Encrypt or Decrypt Multiple Items ......................................................... 913

View Encryption Status for a Single Item ............................................... 913

Filter Data Sources by Encryption Status .............................................. 913
Effects of clickjack protection ......................................................... 920
Disabling clickjack protection ......................................................... 921
HTTP Response Headers ................................................................. 921
Configuring response headers ....................................................... 922
HTTP Strict Transport Security (HSTS) ........................................... 922
Options ......................................................................................... 922
Referrer-Policy .............................................................................. 922
Options ......................................................................................... 923
X-Content-Type-Options ................................................................. 923
Option ......................................................................................... 923
X-XSS-Protection .......................................................................... 924
Option ......................................................................................... 924
Content Security Policy ................................................................. 924
Configure and enable CSP ............................................................... 924
Step 1: Set default directives ......................................................... 925
Step 2: Add additional directives (optional) ...................................... 927
Step 3: Specify report-only directives (optional) .............................. 927
Step 4: Enable CSP on Tableau Server ......................................... 928
Step 5: Run tsm pending-changes apply ....................................... 929
View CSP report ........................................................................... 929
SSL .................................................................................................. 930
Configure SSL for External HTTP Traffic to and from Tableau Server .... 930
SSL certificate requirements .................................................. 931
Configuring SSL for a Cluster ................................................... 932
SSL with multiple gateways ..................................................... 932
Additional configuration information for Tableau Server cluster environments .... 932
Prepare the environment .......................................................... 933
Configure SSL on Tableau Server ............................................. 934
Use the TSM web interface ....................................................... 934
Use the TSM CLI ................................................................... 935
Port redirection and logging ...................................................... 935
Add SSL port to the local firewall ............................................. 936
Change or update SSL certificate .............................................. 936
Example: SSL Certificate - Generate a Key and CSR .................... 937
Steps to generate a key and CSR ............................................... 937
Configure a certificate for multiple domain names ....................... 938
Generate a key ................................................................... 938
Create a certificate signing request to send to a certificate authority .......... 938
Send the CSR to a certificate authority to obtain an SSL certificate .......... 939
Use the key and certificate to configure Tableau Server ................. 939
  For SAN certificates: modify the OpenSSL configuration file .......... 940
Configure SSL for Internal Postgres Communication .................... 942
Use the TSM web interface ....................................................... 942
Use the TSM CLI ................................................................... 943
What the command does .......................................................... 944
Option for repository-ssl enable ........................................... 944
Cluster environments .......................................................... 944
Configure Postgres SSL to Allow Direct Connections from Clients ....... 945
Configure Mutual SSL Authentication ..................................... 946
Use the TSM web interface ................................................... 946
Use the TSM CLI ............................................................... 948
Step 1: Require SSL for external server communication ................... 948
Step 2: Use mutual SSL ....................................................... 948
Additional options for mutual SSL ......................................... 949
Fallback authentication ....................................................... 949
User name mapping .......................................................... 950
Certificate Revocation List (CRL) ........................................ 950
Mapping a Client Certificate to a User During Mutual Authentication .... 950
User-name mapping options ............................................... 951
Change the certificate mapping .......................................... 952
Address user-name mapping ambiguity in multi-domain organizations ... 953
How Mutual SSL Authentication Works .................................. 953
System User, sudo Privileges, and systemd ............................. 955
Privilege separation ......................................................... 955
sudo privileges ............................................................. 955
systemd user service ...................................................... 956
Running systemctl commands .......................................................... 956

Security Hardening Checklist .......................................................... 957

1. Update to the current version ......................................................... 957

2. Configure SSL/TLS with a valid, trusted certificate .......................... 957

3. Disable older versions of TLS ....................................................... 957

4. Configure SSL encryption for internal traffic ................................... 958

5. Enable firewall protection ............................................................ 958

6. Restrict access to the server computer and to important directories .... 959

7. Generate fresh secrets and tokens .................................................. 959

8. Disable services that you’re not using ............................................ 960

   REST API ................................................................................. 960

   JMX Service ........................................................................... 960

9. Verify session lifetime configuration .............................................. 961

10. Configure a server safelist for file-based data sources .................... 961

11. Enable HTTP Strict Transport Security for web browser clients ....... 962

12. Disable Guest access .................................................................... 963

13. Set referrer-policy HTTP header to 'same-origin' ......................... 964

Change List ..................................................................................... 964

Manage Licenses ............................................................................. 965

Licensing Overview ......................................................................... 965

License metric: User-Based or Core-Based ........................................ 965

User-Based licenses ........................................................................ 966
Data Management Add-On ................................................................. 967
Tableau Server Management Add-on ........................................... 968
Tableau Server licenses and site roles after an upgrade ...................... 968
Tableau Server product key activation ............................................. 968
Tableau Server licensing and virtual machines (VMs) ......................... 969

View Server Licenses ........................................................................ 969
Viewing licenses from the Tableau Server web UI .................................. 969
Use the TSM web interface ................................................................. 969
Use the TSM CLI ............................................................................... 970

Refresh Maintenance Date for the Product Key ..................................... 971
Use the TSM web interface ................................................................. 972
Use the TSM CLI ............................................................................... 972

Add Capacity to Tableau Server ......................................................... 973
Use the TSM web interface ................................................................. 973
Use the TSM CLI ............................................................................... 976

Activate Tableau Server Offline - Adding a License ......................... 977
Offline activation overview and prerequisites ..................................... 978
Log in to Tableau Services Manager ................................................ 982
Step 1 - Generate an offline activation request file ............................. 982
Step 2 - Upload the offline activation request to Tableau .................... 983
Step 3 - Initialize or activate your license ........................................... 983
First time license activation ............................................................... 983
Subsequent license activation ............................................................. 984
Automate Licensing Tasks ................................................................. 984
Troubleshoot Licensing ................................................................. 984
Handle an unlicensed server ........................................................... 984
Unlicensed User-Based Server .......................................................... 985
Unlicensed Core-Based Server ........................................................... 985
Troubleshoot user-based licensing ................................................... 985
A user or administrator is unlicensed due to license expiration ............ 986
Server Administrator site role is unchanged when using a Creator license ... 987
 Licenses are not immediately available ........................................... 987
 A user with a Viewer license cannot open Tableau Server or Tableau Online
  workbooks from Tableau Desktop .................................................. 988
Migrate from Core-Based to User-Based Licensing .......................... 988
Prepare for migration to user-based licensing ..................................... 988
Migrate to user-based licensing ....................................................... 989
Use user-based licenses on a server with core-based licensing ............ 989
Example of completing a migration from core-based licensing ............ 990
Login-based License Management .................................................... 991
Use login-based license management .............................................. 991
Step 1: Install Tableau Server .......................................................... 992
Step 2: Enable login-based license management .................................. 992
Step 3: Activate Tableau Desktop ..................................................... 998
Prepare the source and target sites ......................................................... 1033

Tips for importing to a target with fewer users or schedules than the source site ................................................................. 1036

1. Export a site .................................................................................. 1037
2. Generate the import mapping files .................................................. 1038
3. Verify that site settings are mapped correctly .................................. 1038
   To verify mapping files ................................................................ 1039
4. Import the correctly mapped files to the target site ......................... 1040

Mapping file content reference ......................................................... 1040

CSV file name: mappingsDomainMapperForGroups ......................... 1040
CSV file name: mappingsScheduleMapper ...................................... 1041
CSV file name: mappingsSiteMapper ............................................. 1042
CSV file name: mappingsSystemUserNameMapper .......................... 1042
CSV file name: MappingsScheduleRecurrenceMapperWithAutoCreation 1044

Delete Sites ..................................................................................... 1044

Site Availability ............................................................................... 1044
   To activate or suspend a site ....................................................... 1045

Allow Users to Save Revision History ............................................. 1045

Notes ................................................................................................ 1045

Permissions users need to work with revision history ....................... 1046

Enable revision history and set the number of revisions allowed .......... 1046

Clear all revisions .......................................................................... 1047
Security for previewing and restoring workbooks ........................................ 1047
See also ........................................................................................................ 1047

Extract Refresh Schedules ........................................................................... 1047

Before refreshing extracts ........................................................................ 1047
Setting up refresh schedules ................................................................ 1048
Refreshing extracts manually ................................................................ 1049
Refreshing extracts from Tableau Desktop ............................................. 1049

Enable Extract Refresh Scheduling and Failure Notification ...................... 1050
Managing schedules from the server ...................................................... 1051
How refresh failure emails work .......................................................... 1051
How the last successful refresh date is determined ................................. 1052

Create or Modify a Schedule ..................................................................... 1052
To create a new schedule ....................................................................... 1052
To modify an existing schedule ............................................................. 1054
See also .................................................................................................... 1055

Enable Custom Schedules for Subscriptions ............................................. 1055
Enable custom schedules ........................................................................ 1056

How Scheduled Server Tasks are Prioritized ........................................ 1057

Configure Workbook Performance after a Scheduled Refresh .................. 1058
Determine the performance impact ......................................................... 1058
Turn off workbook caching for the server .............................................. 1058
Turn off workbook caching for a site ..................................................... 1059
Stats for Space Usage .......................................................... 1096
Server Disk Space ............................................................... 1098
Desktop License Usage ....................................................... 1100
Who has used Tableau in the last <nn> days? ......................... 1101
What licenses have not been used in the last <nn> days ............ 1102
Desktop License Expiration ................................................. 1102
Create Custom Administrative Views ................................... 1104
Performance ................................................................. 1105
Tableau Server Performance Overview ................................. 1105
General Performance Guidelines ......................................... 1106
Hardware and Software ...................................................... 1106
Configuration ................................................................. 1106
Server Resource Manager (SRM) ......................................... 1107
Performance Monitoring Overview ...................................... 1108
Collect Data with the Tableau Server Repository ..................... 1109
Enable access to the Tableau Server repository ...................... 1109
Connect to the Tableau Server repository ............................. 1110
About the Tableau Server Data Dictionary ............................ 1112
Performance Tuning ......................................................... 1112
Optimize for User Traffic ................................................... 1113
When to optimize for user traffic ........................................ 1113
Slow load times for views .................................................. 1113
Optimize for Query Heavy Environments ........................................... 1126
When to use this configuration ......................................................... 1126
When not to use this configuration .................................................... 1127
Configuration .................................................................................. 1128
General configuration notes ............................................................... 1129
Special configuration notes ................................................................. 1129
Performance Benefits ....................................................................... 1130
Additional Benefits .......................................................................... 1130
When to Add Nodes and Reconfigure ............................................... 1131
Performance Tuning Examples ........................................................... 1132
Performance Recording ..................................................................... 1132
Create a Performance Recording ...................................................... 1132
Start a Performance Recording for a View ......................................... 1133
View a Performance Recording .......................................................... 1134
Interpret a Performance Recording ................................................... 1134
Timeline ......................................................................................... 1135
Events .............................................................................................. 1135
Query ............................................................................................... 1137
Performance Monitoring Tools ......................................................... 1137
Configure Client-Side Rendering ....................................................... 1139
Supported browsers .......................................................................... 1140
Configure the complexity threshold for computers and mobile devices .... 1140
Options ........................................................................................................... 1273

tsm data-access set-saml-delegation disable .............................................. 1274
  Synopsis ........................................................................................................ 1274

  tsm data-access set-saml-delegation enable ................................................ 1274
  Synopsis ........................................................................................................ 1274

  tsm data-access web-data-connectors add .................................................. 1274
  Synopsis ........................................................................................................ 1274
  Options ......................................................................................................... 1274

  tsm data-access web-data-connectors allow ............................................... 1275
  Synopsis ........................................................................................................ 1275
  Options ......................................................................................................... 1275

  tsm data-access web-data-connectors delete .............................................. 1276
  Synopsis ........................................................................................................ 1276
  Options ......................................................................................................... 1276

  tsm data-access web-data-connectors list .................................................... 1276
  Synopsis ........................................................................................................ 1277
  Options ......................................................................................................... 1277
  Global options ............................................................................................. 1277

  tsm email ........................................................................................................... 1278

  tsm email test-smtp-connection ................................................................... 1278
  Synopsis ........................................................................................................ 1278
  Global options ............................................................................................. 1278
tsm licenses get-offline-activation-file ........................................... 1286

Synopsis ................................................................. 1286

Options ........................................................................ 1286

tsml licenses get-offline-deactivation-file ....................................... 1286

Synopsis ................................................................. 1286

Options ........................................................................ 1286

tsml licenses list ............................................................ 1287

Synopsis ................................................................. 1287

tsml licenses refresh .......................................................... 1287

Synopsis ................................................................. 1287

Global options ............................................................... 1287

tsml login ................................................................. 1288

Synopsis ................................................................. 1289

Global options ............................................................... 1289

tsml logout ................................................................. 1290

Synopsis ................................................................. 1290

Global options ............................................................... 1290

tsml maintenance ........................................................... 1291

tsml maintenance backup .................................................... 1292

Synopsis ................................................................. 1292

Options ........................................................................ 1292

Examples ....................................................................... 1294
tsm maintenance cleanup  ................................................................. 1294
Synopsis ......................................................................................... 1295
Options ......................................................................................... 1295
Examples ....................................................................................... 1297

tsm maintenance metadata-services enable ...................................... 1297
Synopsis ......................................................................................... 1298
Option ............................................................................................ 1298

tsm maintenance metadata-services disable ..................................... 1298
Synopsis ......................................................................................... 1298
Option ............................................................................................ 1298

tsm maintenance metadata-services get-status ................................... 1299
Synopsis ......................................................................................... 1299

tsm maintenance reindex-search ...................................................... 1299
Synopsis ......................................................................................... 1299
Option ............................................................................................ 1299

tsm maintenance reset-searchserver .............................................. 1299
Synopsis ......................................................................................... 1300
Option ............................................................................................ 1300

tsm maintenance restore .............................................................. 1300
Synopsis ......................................................................................... 1300
Options ........................................................................................... 1300

tsm maintenance send-logs ............................................................ 1302
<table>
<thead>
<tr>
<th>Command</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Synopsis</td>
<td>1302</td>
</tr>
<tr>
<td>Options</td>
<td>1302</td>
</tr>
<tr>
<td>tsm maintenance validate-resources</td>
<td>1303</td>
</tr>
<tr>
<td>Synopsis</td>
<td>1303</td>
</tr>
<tr>
<td>Options</td>
<td>1303</td>
</tr>
<tr>
<td>tsm maintenance ziplogs</td>
<td>1303</td>
</tr>
<tr>
<td>Synopsis</td>
<td>1304</td>
</tr>
<tr>
<td>Options</td>
<td>1304</td>
</tr>
<tr>
<td>Global options</td>
<td>1306</td>
</tr>
<tr>
<td>tsm pending-changes</td>
<td>1307</td>
</tr>
<tr>
<td>tsm pending-changes apply</td>
<td>1308</td>
</tr>
<tr>
<td>Synopsis</td>
<td>1308</td>
</tr>
<tr>
<td>Options</td>
<td>1308</td>
</tr>
<tr>
<td>tsm pending-changes discard</td>
<td>1309</td>
</tr>
<tr>
<td>Synopsis</td>
<td>1309</td>
</tr>
<tr>
<td>Options</td>
<td>1309</td>
</tr>
<tr>
<td>tsm pending-changes list</td>
<td>1309</td>
</tr>
<tr>
<td>Synopsis</td>
<td>1309</td>
</tr>
<tr>
<td>Options</td>
<td>1310</td>
</tr>
<tr>
<td>Global options</td>
<td>1310</td>
</tr>
<tr>
<td>tsm register</td>
<td>1311</td>
</tr>
<tr>
<td>Synopsis</td>
<td>1311</td>
</tr>
</tbody>
</table>
Synopsis ................................................................. 1319
Options ............................................................... 1319
tsm security external-ssl list ........................................ 1320
Synopsis ................................................................. 1321
Options ............................................................... 1322
tsm security regenerate-internal-tokens .......................... 1321
Synopsis ................................................................. 1322
Options ............................................................... 1322
tsm security repository-ssl disable ................................ 1322
Synopsis ................................................................. 1322
Options ............................................................... 1322
tsm security repository-ssl enable ................................ 1322
Synopsis ................................................................. 1322
Options ............................................................... 1322
tsm security repository-ssl get-certificate-file .................. 1323
Synopsis ................................................................. 1323
Options ............................................................... 1323
tsm security repository-ssl list ...................................... 1323
Synopsis ................................................................. 1323
tsm security vizql-extsvc-ssl disable ............................. 1324
Synopsis ................................................................. 1324
Options ............................................................... 1324
tsm security vizql-extsvc-ssl enable .............................. 1324
Synopsis ................................................................. 1324
Options ............................................................... 1324
<table>
<thead>
<tr>
<th>Command</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Options</td>
<td>1335</td>
</tr>
<tr>
<td>Global options</td>
<td>1335</td>
</tr>
<tr>
<td>tsm start</td>
<td>1336</td>
</tr>
<tr>
<td>Synopsis</td>
<td>1336</td>
</tr>
<tr>
<td>Option</td>
<td>1337</td>
</tr>
<tr>
<td>Global options</td>
<td>1337</td>
</tr>
<tr>
<td>tsm status</td>
<td>1338</td>
</tr>
<tr>
<td>Synopsis</td>
<td>1338</td>
</tr>
<tr>
<td>Options</td>
<td>1338</td>
</tr>
<tr>
<td>Global options</td>
<td>1339</td>
</tr>
<tr>
<td>tsm stop</td>
<td>1340</td>
</tr>
<tr>
<td>Synopsis</td>
<td>1340</td>
</tr>
<tr>
<td>Option</td>
<td>1340</td>
</tr>
<tr>
<td>Global options</td>
<td>1340</td>
</tr>
<tr>
<td>tsm topology</td>
<td>1342</td>
</tr>
<tr>
<td>tsm topology cleanup-coordination-service</td>
<td>1342</td>
</tr>
<tr>
<td>Synopsis</td>
<td>1343</td>
</tr>
<tr>
<td>Option</td>
<td>1343</td>
</tr>
<tr>
<td>tsm topology deploy-coordination-service</td>
<td>1343</td>
</tr>
<tr>
<td>Synopsis</td>
<td>1343</td>
</tr>
<tr>
<td>Options</td>
<td>1343</td>
</tr>
<tr>
<td>tsm topology external-services list</td>
<td>1344</td>
</tr>
</tbody>
</table>
Synopsis ............................................................................................................. 1344
Option ............................................................................................................. 1344
tsm topology external-services repository enable ......................................... 1344
Synopsis ............................................................................................................. 1345
Options ............................................................................................................. 1345
tsm topology external-services repository disable -n nodeN ...................... 1345
Synopsis ............................................................................................................. 1345
Option ............................................................................................................. 1345
Global options .................................................................................................. 1346
tsm topology failover-repository ................................................................. 1347
Synopsis ............................................................................................................. 1347
Options ............................................................................................................. 1347
tsm topology filestore decommission ............................................................. 1348
Synopsis ............................................................................................................. 1348
Options ............................................................................................................. 1348
tsm topology filestore recommission .............................................................. 1349
Synopsis ............................................................................................................. 1349
Options ............................................................................................................. 1349
tsm topology list-nodes ................................................................................... 1349
Synopsis ............................................................................................................. 1349
Options ............................................................................................................. 1350
tsm topology list-ports ................................................................................... 1350
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pass the configuration file to Tableau Server</td>
<td>1412</td>
</tr>
<tr>
<td>See also</td>
<td>1413</td>
</tr>
<tr>
<td>sapHanaSettings Entity</td>
<td>1413</td>
</tr>
<tr>
<td>SAP HANA SAML settings</td>
<td>1414</td>
</tr>
<tr>
<td>Configuration template</td>
<td>1414</td>
</tr>
<tr>
<td>Configuration file reference</td>
<td>1414</td>
</tr>
<tr>
<td>shareProductUsageDataSettings Entity</td>
<td>1415</td>
</tr>
<tr>
<td>Configuration template</td>
<td>1416</td>
</tr>
<tr>
<td>Configuration file reference</td>
<td>1416</td>
</tr>
<tr>
<td>trustedAuthenticationSettings Entity</td>
<td>1417</td>
</tr>
<tr>
<td>Configuration template</td>
<td>1417</td>
</tr>
<tr>
<td>Configuration file reference</td>
<td>1418</td>
</tr>
<tr>
<td>web-data-connector-settings Entity</td>
<td>1420</td>
</tr>
<tr>
<td>Web data connector settings</td>
<td>1421</td>
</tr>
<tr>
<td>Configuration template</td>
<td>1421</td>
</tr>
<tr>
<td>Configuration file reference</td>
<td>1422</td>
</tr>
<tr>
<td>tabcmd</td>
<td>1423</td>
</tr>
<tr>
<td>Install tabcmd</td>
<td>1423</td>
</tr>
<tr>
<td>How to use tabcmd</td>
<td>1428</td>
</tr>
<tr>
<td>Examples</td>
<td>1428</td>
</tr>
<tr>
<td>tabcmd Commands</td>
<td>1429</td>
</tr>
<tr>
<td>addusers group-name</td>
<td>1430</td>
</tr>
</tbody>
</table>
Install fails with "Failed to initialize the instance of the temporary database" 1558

Installation fails on a virtual machine in Parallels ........................... 1558

Tableau Server doesn't start .................................................... 1559

Cannot start Tableau Server after installation ............................... 1559

Cannot create initial administrator account with multiple Active Directory (AD) domains .......................................................... 1560

Fonts .................................................................................... 1560

Support for Asian character sets ................................................. 1561

Initializing Tableau Server ............................................................ 1561

TSM initialization fails because the tableau user account exists but is not a member of the group tableau ........................................ 1561

Error initializing Tableau Server on unsupported system locale .... 1562

Error initializing Tableau Server when en_US.utf8 is not included in locale list ................................................................. 1562

Error: status 10 - initializing Tableau Server when data directory path includes a period ....................................................... 1562

Error initializing Tableau Server after reinstallation .................... 1562

Activating Tableau Server .............................................................. 1563

Tableau Server license activation fails .......................................... 1563

Reindexing Tableau Server Search & Browse ............................... 1563

Problems that can be solved by reindexing Search & Browse .......... 1563

Restarting Tableau Server ............................................................ 1564

Restarting Tableau Server or applying changes fails ................. 1564
Error restarting Tableau Server after adding or configuring a node ........ 1564
Backup/Restore ............................................................................. 1565
File locations ................................................................................. 1566
Changing basefilepath does not change the location of an existing file .... 1566
TSM commands .............................................................................. 1566
TSM command line does not show progress for long-running tasks ........ 1566
Opening Firewall ports .................................................................... 1567
Manually opening firewall ports on Ubuntu and Debian ......................... 1567
OpenID fails on first sign-in attempt ............................................. 1567
Administrative views do not display .............................................. 1567
Changing locale on view .................................................................. 1568
Work with Log Files ....................................................................... 1568
Investigating Tableau Server Issues .............................................. 1568
Server Log File Locations .............................................................. 1570
Tableau Server Log and Configuration File Locations ........................ 1571
  Logs not included in the ziplog archive ...................................... 1571
  Bash script logging ....................................................................... 1572
  System information ....................................................................... 1572
  Log Archive File Locations (unzipped archive) .............................. 1572
Log File Snapshots (Archive Logs) .................................................. 1583
  Use the TSM web interface .......................................................... 1583
  Uploading log snapshots for Tableau Support .............................. 1585
Use the TSM CLI ........................................................................................................1586
Sending log archives to Tableau Support .................................................................1587
Change Logging Levels ............................................................................................1588
Logging Levels ........................................................................................................1588
Change Logging Levels ............................................................................................1589
Reset Logging Levels ..............................................................................................1590
Troubleshoot Tableau Server Install and Upgrade ..................................................1591
General Troubleshooting Steps ................................................................................1591
Common Tableau Server Install Issues .................................................................1592
Installation logs location ........................................................................................1592
Unable to log into TSM or Tableau Server (sign in screen redispays after entering credentials) ..................................................................................1592
Multiple install attempts fail ..................................................................................1592
Install fails due to hardware requirements ............................................................1593
Common Tableau Server Upgrade Issues ..............................................................1593
Upgrade logs location ............................................................................................1593
Upgrade multi-node, initializing additional node fails with "Enter your credentials again" error ..................................................................................1594
Upgrading fails due to lack of disk space ...............................................................1594
Troubleshooting connections to TSM .................................................................1595
Starting Tableau Server ..........................................................................................1595
Tableau Server cannot determine if it fully started ..............................................1595
Tableau Server doesn't start ......................................................... 1595
Reindexing Tableau Server Search & Browse .................................. 1595
Problems that can be solved by rebuilding Search & Browse index ........ 1595
Activating Tableau Server .............................................................. 1596
Tableau Server license activation fails ........................................... 1596
Confirm you can access the licensing server .................................. 1596
Verify the date and time ................................................................. 1596
Force the product key to be read again ........................................... 1598
Send the contents of trusted storage to Tableau Support ................. 1598
Troubleshoot Server Sign in Problems .......................................... 1599
Troubleshooting scenarios ............................................................... 1600
Unable to log into TSM or Tableau Server (sign in screen redisplay after entering credentials) ......................................................... 1600
Troubleshoot Licensing ................................................................. 1600
Handle an unlicensed server ........................................................... 1600
Unlicensed User-Based Server ....................................................... 1601
Unlicensed Core-Based Server ....................................................... 1601
Troubleshoot user-based licensing .................................................. 1601
A user or administrator is unlicensed due to license expiration .......... 1601
Server Administrator site role is unchanged when using a Creator license .. 1603
Licenses are not immediately available ........................................... 1603
A user with a Viewer license cannot open Tableau Server or Tableau 1603
Online workbooks from Tableau Desktop .................................................
Handle an Unlicensed VizQL Server Process ..............................................1603
Tableau Services Manager (TSM) Command Timeout .................................1604
Tableau Services Manager (TSM) Backup Timeout ......................................1605
Cookie Restriction Error ............................................................................1605
Troubleshoot Subscriptions .......................................................................1606
"The view snapshot in this email could not be properly rendered." .............1606
Can't see images in email ............................................................................1606
Can't subscribe .........................................................................................1607
No subscription icon ..................................................................................1607
Receiving invalid or "broken" subscriptions ...............................................1608
Missing PDF attachment ............................................................................1608
Suspension of subscriptions ......................................................................1608
Resume suspended subscriptions ...............................................................1608
Subscriptions not arriving ("Error sending email. Can't send command to
SMTP host.") .........................................................................................1609
Custom scripts not working after upgrade to 8.1 .....................................1609
Custom scripts not working after upgrade to 9.0 .......................................1610
Troubleshoot Performance of Workbooks with Calculations ....................1610

Scenario: You upgraded your Tableau Server recently. You are experi-
encing slow response times and the majority of your extracts have not been
refreshed, so they are still in .tde format. .....................................................1611
Scenario: You upgraded your Tableau Server recently. You are experiencing slow response times and a majority of your extracts have been refreshed and upgraded from .tde to .hyper.

Scenario: You upgraded your Tableau Server a while ago and you are experiencing slow response times. It is not practical for you to roll back using the backup you made prior to upgrade because your workbooks have changed since the upgrade, and the extracts have been upgraded to .hyper format.

Server Administrator Reference

Tableau Server Processes
Licensed processes
Process workflow
Tableau Server Administration Agent
Tableau Server Application Server
Tableau Server Backgrounder Process
Related content
Tableau Server Cache Server
Tableau Server Client File Service
Tableau Server Coordination Service
Configuration for the Coordination Service
The Coordination Service Quorum
Number of Coordination Service instances to use
If you reduce the number of nodes
Viewing Coordination Service Status
Tableau Server Data Engine
Use the TSM web interface ......................................................... 1654
Use the TSM CLI ................................................................. 1655
Tableau Server Repository ..................................................... 1656
Preferred active repository ................................................... 1657
The failoverrepository Command ....................................... 1657
Tableau Server SAML Service ................................................. 1658
Tableau Server Search and Browse ....................................... 1658
Tableau Server Administration Controller Process ............ 1659
Tableau Server TSM Maintenance Services ......................... 1660
Tableau Server VizQL Server ................................................. 1661
Tableau Prep Conductor ....................................................... 1662
  Overview ........................................................................... 1662
Performance and Scale Recommendations ......................... 1663
Topology and Configuration ............................................... 1663
Enabling Tableau Prep Conductor on Tableau Server .......... 1664
Tableau Server Dynamic Topology Changes ......................... 1664
  Example Scenarios .......................................................... 1665
    Making dynamic topology changes .................................. 1665
    Impact of dynamic topology changes ............................... 1665
    Best practices ............................................................. 1666
  Automating dynamic topology changes ............................. 1666
  Additional information .................................................. 1666
<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Server Process Limits</td>
<td>1667</td>
</tr>
<tr>
<td>Tableau Services Manager Ports</td>
<td>1667</td>
</tr>
<tr>
<td>Firewall requirements</td>
<td>1668</td>
</tr>
<tr>
<td>Port assignment</td>
<td>1669</td>
</tr>
<tr>
<td>Dynamic port assignment</td>
<td>1669</td>
</tr>
<tr>
<td>Changing the port range</td>
<td>1670</td>
</tr>
<tr>
<td>Disabling dynamic port assignment</td>
<td>1671</td>
</tr>
<tr>
<td>Manual port assignment</td>
<td>1671</td>
</tr>
<tr>
<td>Configuring ports during installation</td>
<td>1672</td>
</tr>
<tr>
<td>Configuring ports after installation</td>
<td>1673</td>
</tr>
<tr>
<td>Ports that are not dynamically mapped</td>
<td>1674</td>
</tr>
<tr>
<td>Controlling port remapping with initialize-tsm</td>
<td>1675</td>
</tr>
<tr>
<td>Dynamically mapped ports</td>
<td>1676</td>
</tr>
<tr>
<td>Enable the JMX Ports</td>
<td>1682</td>
</tr>
<tr>
<td>How the JMX Ports Are Determined</td>
<td>1683</td>
</tr>
<tr>
<td>Help Output for initialize-tsm Script</td>
<td>1683</td>
</tr>
<tr>
<td>Output</td>
<td>1683</td>
</tr>
<tr>
<td>Related topics</td>
<td>1686</td>
</tr>
<tr>
<td>View Server Version</td>
<td>1687</td>
</tr>
<tr>
<td>Viewing the server version from the Tableau Server web UI</td>
<td>1687</td>
</tr>
<tr>
<td>Viewing the server version and TSM version from the TSM command line</td>
<td>1688</td>
</tr>
<tr>
<td>Short version, long version, and version_code</td>
<td>1688</td>
</tr>
</tbody>
</table>
Finding the long version number .................................................. 1689

Configure Connection to External Services ..................................... 1689

Script errors ................................................................................. 1690

Changing the Identity Store .......................................................... 1691

Warning ....................................................................................... 1692

Methods for restoring content and permissions ............................. 1692

User filters ................................................................................... 1693

User names and the Tableau Identity store .................................... 1693

Method 1: Use site export and import ............................................. 1694

Method 2: Fresh installation—users republish content .................... 1694

Back up, remove, and then reinstall .............................................. 1695

Step 1: Back up Tableau Server .................................................... 1695

Step 2: Remove Tableau Server .................................................... 1695

Step 3: Reinstall Tableau Server with new authentication type ........ 1695

LDAP Configuration Reference ...................................................... 1695

Calculated configKeys ................................................................... 1709

Unsupported configKeys .............................................................. 1710

About Tableau Server Management Add-on ................................. 1710

Server Management Add-on Licensing Requirements .................... 1710

Activating the Server Management Add-on license ....................... 1712

About Tableau Content Migration Tool ......................................... 1714

What is Content Migration Tool? ................................................. 1714
Help and Support ................................................................. 1714
Getting Started with Tableau Content Migration Tool .............. 1714
Pre-Installation ..................................................................... 1714
Installation Requirements .................................................... 1714
Product Compatibility with Tableau Server ............................. 1715
Post-Installation ................................................................... 1715
Migration Plan ..................................................................... 1715
Upgrading Content Migration Tool ........................................ 1715
Install Tableau Content Migration Tool .................................... 1715
Content Migration Tool Installation Requirements .................. 1715
Installing Content Migration Tool .......................................... 1716
Upgrade Tableau Content Migration Tool ............................... 1716
Before Upgrading ................................................................ 1717
Upgrading Content Migration Tool ........................................ 1717
Using Tableau Content Migration Tool .................................... 1717
Migration Plan Overview ...................................................... 1718
Limitations when Migrating Content ...................................... 1718
Encryption Keys .................................................................. 1718
Step 1: Start ...................................................................... 1719
Step 2: Planning ................................................................. 1720
Step 3: Migration ............................................................... 1720
Optional: Roll back ............................................................ 1723
Change Project ...................................................................................................................... 1736
Add Project ............................................................................................................................ 1736
Change Prefix ......................................................................................................................... 1737
Change Suffix .......................................................................................................................... 1738

Step 3: Workbook Transformations ....................................................................................... 1738

Action URL Replacement ...................................................................................................... 1740
Example: ............................................................................................................................... 1740
Set Parameter Value ............................................................................................................ 1741
Remove Images .................................................................................................................... 1741
Remove Tooltip Commands ................................................................................................. 1741
Replace Images .................................................................................................................... 1742
Example: ............................................................................................................................... 1742
Zoom Control Visibility ........................................................................................................ 1742
Web Page URL Replacement ................................................................................................. 1742
Example: ............................................................................................................................... 1743

Step 4: Data Source Transformations .................................................................................... 1743
Set Calculation Formula ...................................................................................................... 1746
Set Connection Info ............................................................................................................. 1746
Set Custom SQL .................................................................................................................... 1747
Remove Extract ..................................................................................................................... 1747

Step 5: Publish Options ........................................................................................................ 1747
Reset Dashboard Selections ................................................................................................. 1748
Destination Project ........................................................................................................1759

Step 3: Data Source Transformations ........................................................................1759

Replace Table/Schema Name .......................................................................................1761

Set Calculation Formula ............................................................................................1761

Set Connection Info ....................................................................................................1762

Set Custom SQL ..........................................................................................................1762

Remove Extract ...........................................................................................................1762

Step 4: Publish Options ...............................................................................................1762

Remove Tags ................................................................................................................1764

Add Tags .......................................................................................................................1765

Apply Extract Refresh Schedules ................................................................................1766

Set Permissions ............................................................................................................1767

Step 5: Continue to Next Step .....................................................................................1768

Migration Plans: Migration Scripts .............................................................................1768

Step 1: Pre-Migration ..................................................................................................1769

Working Directory ......................................................................................................1769

Run ...............................................................................................................................1769

Command Executable ..................................................................................................1769

Command Parameters .................................................................................................1770

Script ............................................................................................................................1770

Step 2: Post-Migration .................................................................................................1770

Working Directory ......................................................................................................1770
Run ................................................................. 1771
Command Executable ........................................... 1771
Command Parameters ........................................... 1771
Script ............................................................... 1771
Step 3: Continue to Next Step .................................. 1771
Migration Plans: Plan Options .................................. 1771
Step 1: Options ..................................................... 1772
Exclude Extract Refreshes ....................................... 1773
Step 2: Version Control ......................................... 1774
Step 3: Save Plan .................................................. 1775
Step 4: Continue to Next Step .................................. 1775
Migrating Workbooks and Data Sources that use Extracts 1776
Changing data connections that use extracts ............ 1777
Option 1: Use Published Data Sources ..................... 1777
Option 2: Remove the Extract During Migration .......... 1778
Option 3: Refresh the Extract After Migration .......... 1778
Consolidate Sites .................................................. 1779
Consolidating sites using the Content Migration Tool ... 1779
Migration Rollback ................................................. 1782
Step 1: Server ....................................................... 1782
Step 2: Migration .................................................. 1784
Step 3: Workbooks ............................................... 1784
Step 4: Published Data Sources ...................................................... 1785

Step 5: Run Rollback ........................................................................ 1786

Using the Tableau Content Migration Tool Console Runner ............... 1788

Run Plan .................................................................................. 1789

Available options: ........................................................................ 1789

Exit codes: .................................................................................. 1790

Show Plan Summary .................................................................. 1790

help ......................................................................................... 1790

version ..................................................................................... 1790

encryption .................................................................................. 1790

improvement ............................................................................. 1790

Examples ................................................................................... 1791

license ....................................................................................... 1791

Examples ................................................................................... 1791

Example: Scripting Migration Plans ............................................. 1792

Using the Tableau Content Migration Tool Command Line Interface .... 1794

migrate ..................................................................................... 1794

help ......................................................................................... 1794

Examples ................................................................................... 1794

license ....................................................................................... 1795

Examples ................................................................................... 1795

update ....................................................................................... 1795
Improvements: .................................................................1807

Enterprise Deployment Tool v1.14 .....................................1807

Improvements: .................................................................1808

Bug Fixes: ...........................................................................1808

Enterprise Deployment Tool v1.13 .....................................1808

Improvements: .................................................................1808

Enterprise Deployment Tool v1.12 .....................................1808

Improvements: .................................................................1808

Enterprise Deployment Tool v1.11 .....................................1809

Improvements: .................................................................1809

Bug Fixes: ...........................................................................1809

Enterprise Deployment Tool v1.10 .....................................1809

Improvements: .................................................................1810

Enterprise Deployment Tool v1.9 .....................................1810

Improvements: .................................................................1810

Bug Fixes: ...........................................................................1810

Enterprise Deployment Tool v1.8 .....................................1810

Improvements: .................................................................1811

Enterprise Deployment Tool v1.7 .....................................1811

Improvements: .................................................................1811

Bug Fixes: ...........................................................................1812

Legacy Compatibility Notes (Power Tools: Deployment) ..........1812
Power Tools: Deployment ................................................................. 1813
Upgrade from Legacy Power Tools: Deployment to Tableau Content
Migration Tool ............................................................................... 1815
Upgrade Process ............................................................................ 1815
Tableau Content Migration Tool Legacy License Key Activation .......... 1815
Serial Key .................................................................................. 1816
Offline Activation ........................................................................ 1816
License File .................................................................................. 1817
Key Management System ................................................................. 1818
Tableau Server local KMS ............................................................... 1818
AWS KMS for encryption at rest ..................................................... 1818
Configure AWS KMS for Tableau Server encrypted extracts .......... 1819
Step 1: Create CMK and set key policy for Tableau Server in AWS .... 1820
Step 2: Collect AWS configuration parameters ............................ 1820
Step 3: Configure Tableau Server for AWS KMS ......................... 1820
Step 4: Enable encryption at rest ................................................ 1821
Step 5: Validate installation .......................................................... 1821
Troubleshoot configuration .......................................................... 1822
Multi-node misconfiguration ......................................................... 1822
Refresh AWS CMK ..................................................................... 1823
Regenerate RMK and MEK on Tableau Server .............................. 1823
Back up and restore with AWS KMS ............................................ 1823
Install and Configure ..............................................................1835
Step 1: Create a configuration file ..............................................1835
Step 2: Install Tableau Server and Configure the External Repository ...1836
Step 3: Complete tsm Initialize ................................................1837
Step 4: Complete the install ......................................................1837
Re-Configure Tableau Server Repository .....................................1837
Move a local Tableau Server Repository to Amazon RDS .................1838
Move the external Tableau Server Repository to the local installation of Tableau Server .........................................................1839
Upgrading your RDS Instance ....................................................1840
Workload Management through Node Roles ..................................1841
Backgrounder Node Roles .......................................................1841
Using Backgrounder Node Roles ..............................................1841
Configuration Options ..........................................................1842
License Requirements ............................................................1843
Considerations .................................................................1844
File Store Node Roles ..........................................................1845
Configuration Options ..........................................................1847
License Requirements ............................................................1848
Use the Data Management Add-on ..........................................1848
License the Data Management Add-on ......................................1849
Tableau Prep Conductor .......................................................1849
Tableau Catalog ................................................................. 1849
How Data Management Add-on licensing works ............................ 1850
User-Based ........................................................................ 1850
Core-Based ......................................................................... 1850
Tableau Prep Conductor ......................................................... 1854
Overview ............................................................................. 1854
Performance and Scale Recommendations .................................... 1855
Topology and Configuration ..................................................... 1855
Enabling Tableau Prep Conductor on Tableau Server ...................... 1856
Enable and Configure Tableau Prep Conductor on Tableau Server .... 1856
Who can do this? .................................................................... 1857
Server Topology ..................................................................... 1857
Next step: ............................................................................ 1859
Step 1 (New Install): Install Tableau Server with Tableau Prep Conductor 1859
Who can do this? .................................................................... 1860
Before you install .................................................................. 1860
Install Tableau Server and enable Tableau Prep Conductor .............. 1860
Configure public gateway settings ........................................... 1860
Enable Tableau Prep Conductor .............................................. 1861
Verify Tableau Prep Conductor is enabled and running .................. 1862
Dedicate a node for Tableau Prep Conductor ............................... 1864
Multi-node installations ......................................................... 1865
Step 3: Create Schedules for Flow Tasks .............................................. 1885
Who can do this? .................................................................................. 1885
Create a new schedule: ................................................................. 1885
Next step ......................................................................................... 1886

Step 4: Safe list Input and Output locations ........................................... 1886
Who can do this? .................................................................................. 1887
How to safe list input and output locations ........................................... 1887
Next step .......................................................................................... 1889

Step 5: Optional Server Configurations .................................................. 1889
Who can do this? ................................................................................... 1889
Set the timeout period for flows ......................................................... 1889
Set the threshold for suspended flow tasks .......................................... 1890

About Tableau Catalog ....................................................................... 1890
How Tableau Catalog works .............................................................. 1891
Key Tableau Catalog terms .................................................................. 1892
License Tableau Catalog ..................................................................... 1892
Enable Tableau Catalog ...................................................................... 1892
Features and functionality .................................................................. 1892
Data discovery .................................................................................... 1893
Curation and trust .............................................................................. 1893
Lineage and impact analysis .............................................................. 1893
Developer resources .......................................................................... 1893
Plan Your Deployment

The articles in this section provide information on planning a Tableau Server deployment.

In addition to consulting the articles in this section, we recommend that you review Tableau Blueprint as a first step in your planning workflow. Tableau Blueprint is a step-by-step guide to becoming a data-driven organization.

- **Get Started with Tableau Server on Linux** .......................................................... 1
- **Server Administrator Overview** ........................................................................... 3
- **Tableau Services Manager Overview** ................................................................. 9
- **Infrastructure Planning** .......................................................................................... 13

Get Started with Tableau Server on Linux


Plan

<table>
<thead>
<tr>
<th>Tableau Blueprint</th>
<th>Before you install...</th>
</tr>
</thead>
<tbody>
<tr>
<td>Server Administrator Overview</td>
<td>Security</td>
</tr>
<tr>
<td>Infrastructure Planning</td>
<td>Planning a Site</td>
</tr>
<tr>
<td>Tableau Services Manager Overview</td>
<td></td>
</tr>
</tbody>
</table>

Deploy

<table>
<thead>
<tr>
<th>Install and Configure Tableau Server</th>
<th>Install Drivers</th>
</tr>
</thead>
<tbody>
<tr>
<td>Distributed and High Availability Tableau Server Installations</td>
<td>Install Tableau Server in the Cloud</td>
</tr>
<tr>
<td>---------------------------------------------------------------</td>
<td>-------------------------------------</td>
</tr>
<tr>
<td>Jump-start Installation</td>
<td>Migrate Tableau Server from Windows to Linux</td>
</tr>
<tr>
<td><strong>Tableau Mobile Deployment Guide</strong></td>
<td>Tableau Desktop and Tableau Prep Deployment Guide</td>
</tr>
</tbody>
</table>

**Manage Server**

<table>
<thead>
<tr>
<th>tsm Command Line Reference</th>
<th>Extract Refresh Schedules</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add or Update Sites</td>
<td>Add Users to Tableau Server</td>
</tr>
</tbody>
</table>

**Manage Sites**

<table>
<thead>
<tr>
<th>What is a site</th>
<th>Manage Users and Groups</th>
</tr>
</thead>
<tbody>
<tr>
<td>Manage Content Access</td>
<td>Manage Data</td>
</tr>
</tbody>
</table>

**Automate and Extend**

<table>
<thead>
<tr>
<th>Tableau Developer Program</th>
<th>Tableau Tools and Samples on GitHub</th>
</tr>
</thead>
<tbody>
<tr>
<td>REST API</td>
<td>tsm Command Line Reference</td>
</tr>
<tr>
<td>Web Data Connector (WDC)</td>
<td>Hyper API</td>
</tr>
</tbody>
</table>

Manage Dashboard Extensions in Tableau Server
Server Administrator Overview

Tableau Server on Linux integrates with a number of components in your IT infrastructure to provide a unique self-service data analytics culture for your users. It’s important that you, as a server administrator, understand how Tableau Server fits into your IT infrastructure.

The topics in this section provide information on planning, deploying, tuning, and managing Tableau Server.

If you are new to Tableau Server, and you want to deploy it in your organization, we encourage you to deploy Tableau Server as a single server in a test environment first. The easiest way to do a single-server installation and to understand the essential requirements is to follow the steps in Everybody’s Install Guide.

If you are deploying Tableau Server as part of a broader effort to transform your organization into a data-driven culture, see Tableau Blueprint. Tableau Blueprint is a step-by-step guide to becoming a data-driven organization, whether your organization is new to modern, self-service analytics or you’ve already deployed and need to broaden, deepen, and scale the use of data.

This topic provides a brief overview of how to think about Tableau Server and how it interacts with your existing IT infrastructure.


Architectural overview

Tableau Server is a collection of processes that work together to provide a full self-service analytic platform for your users. The following diagram shows a high-level architectural view of Tableau Server.
Multiple server processes (shown in blue above) work together to provide services at various tiers. The Gateway process is the component that redirects traffic from all Tableau clients to the available server nodes in a cluster.

Data Services is a logical grouping of services that provide data freshness, shared metadata management, governed data sources, and in-memory data. The underlying processes that power Data Services are the Backgrounder, Data Server and Data Engine processes.

Analytics Services, composed of the VizQL and Cache Server processes, provide user-facing visualization and analytics services and caching services.

Sharing and Collaboration, and Content Management Service are powered by the Application Server process. Core Tableau Server functionality such as user login, content management (projects, sites, permissioning, etc.) and administration activities are provided by the Application Server process.

All of the above services use and rely on the Repository process, which contains structured relational data like metadata, permissions, workbooks, data extracts, user info, and other data. The File Store process enables data extract file redundancy across the cluster and ensures extracts are locally available on all cluster nodes. Under heavier loads, extract files are available locally across the cluster for faster processing and rendering.
Tableau’s architecture is flexible, allowing you to run the platform just about anywhere. You can install Tableau Server on-premises, in your private cloud or data center, on Amazon EC2, on Google Cloud Platform, or on MS Azure. Tableau analytics platform can also run atop virtualization platforms. We recommend you follow the best practices for each virtualization platform to ensure the best performance from Tableau Server.

Tableau and your data

When you install Tableau Server into your organization, it becomes a core component of the analytics pipeline to the data your users need. It’s important to understand how Tableau Server interacts with your business data. Specifically, Tableau Server can store extracts of data in your organization. It can also connect to live data sources. How you choose to provide the data to your Tableau users is informed by a number of variables: data source type, user scenario, performance and access requirements, and infrastructure conditions.

Tableau Server has not been architected as a data warehouse server where static, native data files are housed. In fact, using Tableau Server as a traditional data warehouse is a poor use of your investment. Rather, when it comes to data storage, we recommend hosting optimized data extracts on Tableau Server. While a data extract is often a subset of a larger data source in your organization, you can also create extracts for data sources that are overtaxed during work hours by scheduling the extract refresh for off-hours.

Extracts are also useful for modeling data or to enable highly-performant visualization authoring. For example, to improve visualization authoring and interaction performance you may optimize extracts by filtering the source data to the essential fields for a given department or project.

Tableau Server also provides direct, authorized access to live data sources, allowing users to build and run complex filtered queries against a variety of connected data sources. For this scenario, Tableau requires highly performant network access to the data sources in your organization and to those in the cloud. Tableau Server and the target data sources also need to be properly sized to handle the processing load required by high-volume, complex data operations.
User access

Tableau Server is also a web-based collaboration platform, where users connect to share, view, and interact with data visualizations and data sources from a variety of devices. This means that Tableau Server must be accessible to Tableau users within your local protected network. You can also extend access to data visualizations to desktop, mobile, and authenticated web users outside your organization.

Tableau Server integrates with the following user authentication solutions: Active Directory, SAML, OpenId, and Kerberos.

Where should I install Tableau Server in my network?

Because of the highly-sensitive nature of most data that organizations manage with Tableau Server, and because Tableau Server requires access to internal data stores, Tableau Server must be run inside a protected network. Authenticated access from the internet is configured to connect to Tableau Server through a reverse proxy or a VPN solution.

Some organizations embed Tableau views in public webpages, or, for internal users, on generic web servers on the internal network.

Tableau Server can be configured to support such scenarios with either authenticated or anonymous access. For authorized access, where users can only view underlying data to which they have permission, you can configure trusted tickets with a generic web server.
this scenario, Tableau Server authorizes access to the underlying data in an embedded view. This scheme enables you to host interactive data visualizations on a web server in a DMZ or outside the protected network.

Anonymous access to embedded Tableau views requires that you enable "guest user" for Tableau Server. Guest user also requires that you license Tableau Server according to the number of cores you are running, rather than a named-user (interactor) model.

Sizing and scalability

Depending on the size and data usage in your organization, you can scale Tableau Server up or out. As you scale your server, you can also selectively allocate resources to meet your data needs and user needs.

When you scale up Tableau Server, you add hardware resources to a single server. For example, you might increase the memory and processing power of the computer running Tableau Server.

When you scale out Tableau Server, you add computers (or nodes). To create a highly available deployment with failover, you need at least three nodes. For example, you might run most CPU-intensive server processes on two nodes and use the third node for the gateway and coordination controller services.

Whether you scale up or scale out, you can selectively allocate resources by configuring the number and type of server processes that run. If your organization has a lot of data and creates a lot of data extracts, you can increase the number of processes that are dedicated to refreshing and storing extracts. Alternatively, if your organization wants to optimize for heavy user loads, you can increase the number of processes dedicated to responding to user requests. Additionally, you can integrate Tableau Server into industry-standard network load balancers to further optimize your server for user requests.
Tableau Server management model

Tableau Server has been designed to support a management scheme with two high-level administrators: server administrator and site administrator. In small organizations, these roles may be assumed by the same person or team, but in larger organizations, the roles often diverge.

In this model, server administrators are IT professionals who maintain and deploy heterogeneous server solutions. Essential areas for server administrators may include networking, hardware tuning and maintenance, security and access, and managing users and directory services. The tools and documentation that we deliver with Tableau Server for the server administrator support these core server IT areas.

Site administrator, on the other hand, is an administrative role specific to Tableau Server or Tableau Online deployments. The Tableau site administrator is fundamentally concerned with data content. The site administrator manages users and their access to projects, workbooks, and data sources. To learn about sites and how to plan your deployment for them, see What is a site

Management tools

Tableau Server includes a number of toolsets for managing the system:

- **Tableau Server administrator page:** This is the web-based administrative site that is installed on each Tableau Server instance. Tasks performed on the administrator page are day-to-day tasks for both server and site administrators. Server-related tasks include creating sites and site administrator accounts, optionally importing users, setting up synchronization with directory services, setting up extract refresh schedules, monitoring server performance and usage, and other global settings.

  Site-related tasks include managing content and assign permissions, running extract refreshes, create groups and projects, monitoring site activity, optionally adding users, and other content-related tasks.

  See Navigate the Admin Areas of the Tableau Web Environment.
Permissions required for the Tableau Server administrator page are based on site roles. The site roles are generated and managed by Tableau Server.

- **tsm Command Line Reference** - This is the primary interface for server-wide configurations. The configurations made with TSM CLI are rarely revisited after initial configuration. For example: SSL, subscriptions, data caching, service account, SMTP alerting, user authentication, and single-sign on configuration are all performed with TSM CLI.
- **tabcmd**: You can use the tabcmd command-line utility on a Windows or Linux computer to create scripts to automate administrative tasks on your Tableau Server sites. For example, use tabcmd for creating or deleting users, projects, and groups.
- **REST API**: With the Tableau Server REST API you can manage and change Tableau Server resources programmatically, via HTTP. The API gives you simple access to the functionality behind the data sources, projects, workbooks, site users, and sites on a Tableau server. You can use this access to create your own custom applications or to script interactions with Tableau Server resources.

Security

As an application server connecting to data that may be highly-sensitive, Tableau Server supports and implements a number of industry security standards. Our server admin documentation includes best practices and implementation for user authentication, authorization, data security, and network security. While our default installation is secure by design, we also recommend following the security hardening checklist to further lock down your deployment.

For more information about security audit compliance, vulnerability reporting, and other security resources, visit [http://www.tableau.com/security](http://www.tableau.com/security).

Tableau Services Manager Overview

This article provides an overview of Tableau Services Manager (TSM), which you can use to configure and administer Tableau Server. The TSM CLI was introduced with Tableau Server on Linux, version 10.5. Beginning with version 2018.2, the TSM Web UI is available.
Functionality

TSM gives server administrators command-line and web-based options for configuring and maintaining Tableau Server, including performing administrative tasks like backing up server data, restoring backups, creating log archives, and managing multi-node clusters. For example, you use TSM to perform the following tasks:

- Initial configuration of Tableau Server after installation
- Ongoing configuration management, including editing settings and changing the server topology
- Running administrative tasks such as backup, restore, and ziplogs

For administrators familiar with earlier versions of Tableau Server, TSM replaces the following tools from previous versions of Tableau Server:

- Tableau Server Configuration utility
- tabadmin command line utility
- Tableau Server Monitor

Components

TSM consists of services (called processes in this documentation) and clients. TSM processes are administrative services which manage Tableau Server processes. TSM processes run continuously after TSM is initialized, even when the rest of Tableau Server is offline.

TSM processes that run, even when Tableau Server is stopped include:

- Administration Agent
- Administration Controller
- Client File Service
- Coordination Service (based on Apache Zookeeper™)
For more information about TSM processes and Tableau Server processes, see Tableau Server Processes.

**TSM Authentication**

Whether you use the TSM Web UI, the command line interface, or the TSM API, you need to authenticate to Tableau Server before you can perform administrative tasks. This user account is distinct from Tableau Server user accounts, including Tableau Server administrators and site administrators.

TSM delegates authentication of users to the underlying operating system. On Linux, this means that authentication is handled using Pluggable Authentication Modules (PAM). PAM is the standard on all Linux distributions on which Tableau Server is supported. If your organization has configured PAM to authenticate with your directory service (Active Directory, LDAP), then you can authorize any user from that directory service to access TSM. In this scenario, any authenticated PAM user that is a member of the tsmadmin group is authorized to access TSM.

In the 2019.1 release, TSM authentication process uses PAM directly and then falls back to an authentication scheme using `substitute user (su)` if PAM fails or is not configured with a directory service. If PAM is not configured with a directory service then local accounts must be managed on the Linux computer. In these cases, TSM will use the `su` method of authentication: passing the user-provided credentials to run the `true` command in the `/bin` directory. If that command succeeds, then authentication is verified. Therefore, if the user is a member of the tsmadmin group, then the authenticated user is granted access to TSM.

**Custom PAM service definition**

TSM uses the standard PAM `login` service to authenticate. You can further customize TSM authentication behavior by creating a `tableau` PAM service file in `/etc/pam.d`. If this file exists, then it will be consulted instead of the PAM login service.
Connecting TSM clients

As a security measure, you can only connect to TSM with clients (CLI, Web UI, Rest API) over HTTPS. This is because TSM allows you to perform administrative tasks and to connect to TSM from other computers.

When you are connecting with a TSM client, you must connect to the Tableau Server instance running the TSM Administration Controller service.

**Note:** The TSM CLI tool does not require admin credentials in some scenarios. Specifically, if the account you are logged in as is a member of the TSM-authorized group, you do not need to provide credentials to run commands when running tsm CLI locally. For more information, see Authenticating with tsm CLI.

TSM HTTPS connections rely on a self-signed certificate generated by the Tableau Server installer. This certificate is the Tableau installation CA certificate that signs the SSL certificates Tableau creates for encrypting traffic over HTTP. The Tableau installation CA certificate must be trusted by the systems connecting to TSM Administration Controller.

The TSM CLI client validates certificate trust from a different store than the TSM Web UI uses. The TSM CLI client refers to the trusted store in the local Java keystore to validate trust for CA certificates. Since the TSM Web UI must establish connection with a web browser, trust is validated with the operating system's trusted keystore. The difference in how CA certificates are stored determines different trust configuration scenarios as outlined here:

- For TSM CLI communications on Tableau Server, the certificate trust is configured by default as part of the installation, node bootstrap, and upgrade processes. The Tableau installation CA certificate is added to the trusted store in the Java keystore. This allows you to access TSM using the CLI from any computer in the cluster without additional configuration. However, when accessing TSM Web UI, the browser will prompt you to trust the host running TSM Administration Controller service.
- For TSM CLI connections from remote computers, you will be prompted to trust the Tableau installation CA certificate the first time you connect to the Tableau Server...
Running TSM Administration Controller. You can choose to trust the CA certificate, in which case you will not be prompted again on that computer until the certificate expires (default is 3 years). Or you can connect with a one-time trust by running your TSM command with the `--trust-admin-controller-cert` flag.

- For TSM Web UI connections from remote computers, the browser will prompt you to trust the host running TSM Administration Controller service.

Infrastructure Planning

Before you install Tableau Server, you should review the disk requirements, recommended configurations, user accounts, security, and networking requirements.

Before you install...

**Note:** You can find additional information about technical specifications for Tableau Server on the Tableau web site, [here](#).

This topic includes requirements and recommendations that you must consider before you install Tableau Server into a production environment.

If you want to install a single server, or if you want to do a minimal installation for test purposes, refer to our single-server installation guide, [Everybody's Install Guide](#).

If you are deploying Tableau Server in a distributed cluster, review Distributed Requirements in addition to the requirements and recommendations described in this topic.

If you are migrating from Tableau Server on Windows to Tableau Server on Linux, see Migrate Tableau Server from Windows to Linux.

**Hardware recommendations**

The following list describes the minimum hardware recommendations for a single-node installation of Tableau Server:
• 8 core, 2.0 GHz or higher processor

• 64-bit processor architecture

• 32 GB memory

• 50 GB disk space available, with a minimum of 15 GB allocated to the /opt directory, and the remainder allocated to the /var directory for data storage. See the section, Data directory, below.

**Important:** The disk space requirement cannot be checked until you initialize TSM. If you don’t have enough space, you won’t be told this until after you install the Tableau Server package.

To see the full list of recommendations and to see the minimum requirements, see Minimum Hardware Requirements and Recommendations for Tableau Server.

For public cloud deployments on Amazon Web Services and Google Cloud Platform, their “vCPU” is actually a CPU hyper-thread, and not a full CPU core. When sizing cloud instances, you will need twice as many vCPU as the Tableau Server CPU core requirements given (4 vCPU for a minimum trial installation, 16 vCPU for a recommended installation).

**Operating system requirements**

The following distributions of Linux are supported:

• Red Hat Enterprise Linux (RHEL) 7.3 and higher, CentOS 7.3 and higher, Oracle Linux 7.3 and higher, and Amazon Linux 2.

  These distributions are collectively referred to in this documentation as RHEL-like.

  RHEL 8 is not supported.

• The latest versions of Ubuntu 16.04 LTS and 18.04 LTS.
Non-LTS releases are not supported. Ubuntu version 17.04 is not supported.

- Debian 9

Previous versions of CentOS and Ubuntu are not supported because Tableau Server requires systemd for process management. The version of the installer with the file suffix, .deb, installs on both Ubuntu and Debian distributions.

Custom kernels are not supported.

In a multi-node installation of Tableau Server, all of the computer nodes where you are installing Tableau must run Linux and the same distribution of Linux.

**Note:** All nodes in a multi-node cluster must have the same type of operating system. You cannot install a multi-node instance of Tableau Server on a combination of Linux and Windows nodes.

**Installation directory**

The core Tableau Server bits are installed in the /opt directory by default. You can change the install directory on RHEL-like systems, but cannot change this on Ubuntu. You cannot specify a symbolic link or a directory location on a Network File System (NFS) volume. The directory where you install Tableau Server must have at least 15 GB of free disk space. If you attempt to install Tableau Server on a computer that does not have enough space, the Tableau Server package will install, but you will be unable to continue with setup.

**Data directory**

By default, Tableau Server will create a data directory for all content and extracts that are managed by Tableau. The directory is created at /var/opt/tableau/tableau_server.

You can specify a different directory for data (extract) storage during installation. If you plan to use a different directory, do not create the directory. Instead, let Tableau Server setup
create the directory. The data directory requires specific permissions that are set during the installation process.

To change the data directory, you must pass a flag along with the data directory path when you run the initialize-tsm script. See Help Output for initialize-tsm Script.

If you are changing the default data directory:

- Do not specify a symbolic link or a data directory location on a Network File System (NFS) volume.
- Do not specify a data directory location with a path that includes a period or space. If there is a period or space in the path, initialization will fail.
- The data directory must be installed into a different directory than the installation directory.

**Important:** You cannot change the data directory location after you’ve run `initialize-tsm`. The data directory location will persist for the life of the deployment, including subsequent upgrades.

**Tableau Prep Conductor**

Tableau Prep Conductor is one of the process on Tableau Server. It runs a flow, checks connection credentials, and sends alerts if a flow fails. Tableau Prep Conductor leverages the scheduling and tracking functionality of Tableau Server so you can automate running flows to update the flow output instead of logging into Tableau Prep Builder to manually run individual flows as your data changes.

Tableau Prep Conductor is licensed separately and is available through the Data Management Add-on license. For more information on Tableau Prep Conductor licensing, see [Licensing Tableau Prep Conductor](#).

We recommend you enable Tableau Prep Conductor on a dedicated node. For more information:
If you are installing a new Tableau Server, see Installing Tableau Server with Prep Conductor.

If you are adding Tableau Prep Conductor to an existing installation of Tableau Server, see Add Tableau Prep Conductor to your Tableau Server Installation.

Additional requirements

Make sure that your environment also meets the following additional requirements:

Hostname

- Tableau Server must be able to resolve the hostname to an IP address either using the domain name server (DNS) or with a local host file on the computer running Tableau Server. By default, host files are stored at /etc/hosts.
- The hostname of the server must not change after you start Tableau Services Manager during the setup process. For example, this might happen if you use the cloud-init package to initialize a virtual machine, and you install Tableau Server on that virtual machine.
- Hostnames that include underscores (_) are not supported by Tableau Server on Linux.

Static IP address

The computer where you install Tableau Server must have a static IPv4 or IPv6 address.

Database drivers

To connect to specific data sources, the computer where you install Tableau Server must have the correct database drivers installed. For more information, see Database Drivers.

Available ports

TSM and Tableau Server each require an available TCP port in order for you to access them. TSM defaults to port 8850, and the Tableau Server Gateway service defaults to port 80. We strongly recommend that you ensure that both port 8850 and 80 are not in use on your system before installing Tableau Server. If those ports are not available, the TSM and
gateway ports may be dynamically remapped to different port numbers, and there is currently no interface for displaying which port they have been remapped to.

See Tableau Services Manager Ports.

**Local firewall configuration**

If you are running a firewall on the computer where you will be installing Tableau Server, then you will need to open the following default ports for Tableau Server traffic. All port numbers, except 443 can be changed.

<table>
<thead>
<tr>
<th>Port</th>
<th>TCP/UDP</th>
<th>Used by ...</th>
<th>TYPE OF INSTALLATION</th>
<th>Distributed / High Availability</th>
</tr>
</thead>
<tbody>
<tr>
<td>80</td>
<td>TCP</td>
<td>Gateway</td>
<td>All</td>
<td>X</td>
</tr>
<tr>
<td>443</td>
<td>TCP</td>
<td>SSL. When Tableau Server is configured for SSL, the application server redirects requests to this port. Do not change this port.</td>
<td>All</td>
<td>X</td>
</tr>
<tr>
<td>8850</td>
<td>TCP</td>
<td>Tableau Services Manager.</td>
<td>All</td>
<td>X</td>
</tr>
<tr>
<td>8060</td>
<td>TCP</td>
<td>PostgreSQL database.</td>
<td>All</td>
<td>X</td>
</tr>
<tr>
<td>8061</td>
<td>TCP</td>
<td>PostgreSQL backup verification port</td>
<td>All</td>
<td>X</td>
</tr>
<tr>
<td>8000-9000</td>
<td>TCP</td>
<td>Range of ports reserved by default for dynamic mapping of Tableau processes</td>
<td>All</td>
<td>X</td>
</tr>
<tr>
<td>27000-27009</td>
<td>TCP</td>
<td>Range of ports used by Tableau Server for License service. This range must be open on the node running the License service and accessible from other nodes. By default, the initial node runs the License</td>
<td>All</td>
<td>X</td>
</tr>
</tbody>
</table>
See Tableau Services Manager Ports and Configure Local Firewall.

**System user and groups**

Tableau Server on Linux uses one unprivileged user, and two groups for proper operation. Tableau will create the default account and groups during setup. Alternatively, you can specify existing accounts. See System user and groups.

**Sudo and root access**

All installation tasks and administrative tasks for Tableau Server must be run as root. Often this is accomplished using the sudo command, but running the commands directly as the root user is also possible.

To install Tableau Server with the root account, you must specify a user account during installation. The account will be used for managing TSM. Specify the account by running the initialize-tsm script with the \

\n
**Account password**

The user account that you use to install and administer Tableau Server must be able to authenticate with a password. That is, the user must not use another means of authenticating (for example public key authentication).

If the account you are using to install and initialize Tableau Server does not have a password, you can set one using the `passwd` command:
Port access requirements

If you want to install Tableau Server remotely, for example by means of SSH, ensure that the following ports are open:

- **8850.** The port used for the Tableau Services Manager (TSM) web interface. You can use this interface to configure Tableau Server.
- **80.** The port used for the main Tableau Server web interface.

The Tableau Server installer attempts to open these ports during the installation process, but it can only open these ports for the `firewalld` firewall. If your computer runs another firewall, you must open the ports before you install.

Virtual Container environments

Installation of Tableau Server in virtual container environments such as Docker have not been tested and are not supported. Tableau Server on Linux will not function as expected if installed in these environments.

Package requirements

**Systemd**

Tableau Server requires `systemd` to manage services. This package is installed by default on CentOS 7 and Ubuntu 16. If you decide to test Tableau Server on a modified version of these distributions, you can run the following command to confirm that `systemd` is installed:

```
whereis systemd
```

If `systemd` is installed, the installation location is displayed. For example, you might see the following output:
If you have systemd installed but the Tableau installer is failing requirements checks for systemd, it's likely that systemd is not running. To verify that systemd is running, run the following command:

```bash
ls /run/systemd
```

The output will be a list of files and directories. If systemd is running, the output will include `system`. If `system` is not in the output, then systemd is not running.

*Continue to the next step: Install and Configure Tableau Server.*

## Disk Space Requirements

In general, when estimating the amount of additional disk space to allocate for Tableau Server for day-to-day usage, you must consider whether or not extracts will be published to Tableau Server, and consider the number of workbooks that you expect to publish to Tableau Server. If you anticipate using extracts, Tableau recommends that you begin with a few hundred gigabytes (GB). If you do not anticipate using extracts, you may only need around 50 GB to fulfill your usage needs. To setup drive space alerts, see Configure Server Event Notification.

Looking for Tableau Server on Windows? See Disk Space Requirements.

Here are the factors that affect disk space requirements:

*Publishing extracts to Tableau Server*

Consider the number of extracts that will be published to Tableau Server and the size of each extract. Test your needs by publishing several extracts to Tableau Server, and then checking the disk space used. You can use this amount of disk space to help you figure out
how many extracts will be published to Tableau Server over time as well as how each existing extract will increase in size.

Refreshing extracts

Consider the space needed by the temp directory during an extract refresh. The temp directory, which is where an extract is stored to during a refresh, may require up to three times the final file size of the extract.

While there may be some cases where .hyper extracts become bigger after an extract upgrade, there are other cases when .hyper extracts become smaller after an extract upgrade. In general, the overall disk size requirements should be the same for .hyper extracts as it was for .tde extracts.

Creating many workbooks

If using workbooks, consider the number of workbooks that will be published to Tableau Server. Individual workbooks tend to take up a small amount of disk space. However, if you anticipate thousands of workbooks being published, you may want to allocate additional disk space to accommodate those workbooks.

Logging

To assist with daily management and troubleshooting, Tableau Server creates log files as a part of its normal operations. Depending on the level at which the logging is configured, it can significantly impact the amount of disk space necessary on the Tableau Server computer.

Backup and restore processes

Backup disk space requirements

You need sufficient free disk space to restore from backup, that is in all cases 15% of the disk size. In some cases it could be even greater. During backups, the background tasks for cleaning up old extracts are temporarily paused. This means that, for the duration of the backup, extract refreshes will leave extra files in place, adding to disk space usage. If your
backup takes a long time, or if your organization uses many extracts that are regularly updated, this can result in a significant amount of temporary disk space usage. These old files will be removed after the backup is complete.

- On nodes with the repository, you need additional free space equal to at least twice the size of the repository data. To obtain an estimate of the repository data, check the size of `<data directory>/pgsql/data/base directory`.

- On nodes with the file store, you need additional free space equal to at least the size of the file store data. To obtain an estimate of file store data (extracts, flows, etc.), check the size of `<data directory>/dataengine directory`.

- On nodes with the Administration Controller or TSM Controller (usually the initial node unless you have had to recover from an initial node failure), you need additional free space equal to at least the size of the repository data plus the size of the file store data.

If all of the above are on the same node, you can use the sum of the sizes to determine the amount of disk space required.

### Restore disk space requirements

You must have adequate disk space for the database restore process to run successfully.

To restore Tableau Server:

- On controller nodes, you need free space equal to at least the size of the backup archive.

- On repository nodes, you need free space equal to at least twice the size of the repository data in the backup archive.

- On file store nodes, you need free space equal to at least twice the size of the dataengine folder in the backup archive.
Recommended Baseline Configurations

Determining the topology (number of nodes, number of Tableau Server processes) of your Tableau Server deployment requires you to consider these variables: your environment, sources of data and management to provide self-service data access, workload, and usage. However you may not have enough information about these variables when you deploy Tableau Server for the first time. This topic describes three baseline architectures that can be used as starting points for your Tableau Server installations.

Hardware Recommendations

The hardware recommendations seen below are based on the hardware that the Tableau team uses to test Tableau Server scalability. We suggest that you use these recommendations as starting points for your production deployments. For proof-of-concept deployments, see Minimum Hardware Requirements and Recommendations for Tableau Server.

<table>
<thead>
<tr>
<th>Install Type</th>
<th>Processor</th>
<th>CPU</th>
<th>RAM</th>
<th>Free Disk Space</th>
</tr>
</thead>
<tbody>
<tr>
<td>Single Server</td>
<td>64-bit</td>
<td>8 physical cores</td>
<td>64 GB</td>
<td>500 GB - 1 TB</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(16vCPUs), 2.0 GHz or higher</td>
<td>(8GB/physical Core)</td>
<td></td>
</tr>
</tbody>
</table>

If you are adding Tableau Prep Conductor to your Tableau Server installation, it is recommended to add a second node and dedicate this to running Tableau Server Prep Conductor. This node should have a minimum or 4 physical cores (8 vCPUs), and 16 GB of RAM.

Multi-node and enterprise deployments

Nodes must meet or exceed the minimum hardware recommendations, except in the following scenarios where a node can be configured with 4 physical cores (8 vCPUs):

- Dedicated node for Backgrounder.
- Dedicated node for Tableau Prep Conductor.
- Move all the licensed process from the initial node to additional nodes.
**Note:** For deployments using virtual machines, Tableau recommends dedicated CPU affinity. If you are running Tableau Server in a virtual environment, use your VM host’s best practices for vCPU allocation in relation to the number of physical CPU cores on the VM host. Typically 2 vCPUs = 1 physical core for Tableau Server. For example, for AWS installations, the 4 core minimum recommendation is equivalent of 8 AWS vCPUs. Similarly, follow the best practices provided by your virtual infrastructure provider to make sure Tableau Server has access to the appropriate compute, memory, and data resources. If you are installing Tableau Server in a virtual environment or a cloud-based deployment, see Virtual Machines and Public Cloud Deployments section later in this topic.

### Estimating Disk Space

There are several factors that affect disk space requirements, including whether or not you will be publishing extracts, flows, and the number of workbooks to Tableau Server. For more information see Disk Space Requirements.

### Baseline Configurations

- Single Server Installations
- Two Node Installation - Specialized for extract heavy environments
- Two Node Installation - Specialized for flow environments
- High Availability Installations (HA)

#### Single Server Installations

**Recommendations**

We recommend using a single machine to install your Tableau Server for initial deployments with limited usage and are not mission critical. Single Server installations can also be expanded to multi-node installation as your workloads grow.

Here are some instances when a single server installation may not be right for you:
If your system is considered mission critical and needs to be highly available. High availability is about minimizing the system downtime. It is achieved by eliminating single points of failure, and having a reliable failover mechanism. Tableau Server requires a minimum of a three-node configuration to provide redundancy and eliminate the single points of failure. This is one of the primary reasons to move to a multi-node configuration.

If you have a lot of active users and a lot of extract refreshes, the two types of loads may be competing for the same resources on the machine. In such a scenario, a single server configuration may not be the right option as you may need additional specialized nodes to isolate the difference workloads.

**Note:** Active users represent the interactive, concurrent requests made to Tableau Server, including consuming dashboards on a laptop or mobile device, web authoring, and connecting to and querying Published Data Sources.
Stand-alone Tableau Server node

- Gateway
- Search & Browse
- Application Server (VizPortal) x2
- VizQL Server x2
- Data Server x2
- Ask Data
- Elastic Server
- Cache Server x2
- Backgrounder x2
- Interactive Microservice container
- Non-Interactive Microservice Container

- Data Engine
- File Store
- Repository
- Licensing
- Cluster Controller
- Coordination Service
Multi-Node Installations

Running Tableau Server on more than one machine is called a multi-node installation, or a cluster. There are various reasons why you might want to have a multi-node installation. For example, you may have heavy extract environments which can mean dedicating some hardware resources to Backgrounder process. For systems that have high availability requirements, you need a multi-node environment that has at least three nodes.

Two Node Installation - Specialized for extract heavy environments

Recommendations

Start with a two node configuration when the following conditions apply to you:
• **Extract heavy environment**: Majority of your data sources are extracts. Having just a few, extremely large extracts could put your deployment in this category, as would having very many small extracts.

• **Frequent extract refreshes**: Refreshing an extract is a CPU-intensive task. Deployments where extracts are frequently refreshed (for example, several times a day during business hours) are often helped by more emphasis on the background process, which handles refresh tasks.

**Important**: Two-node configurations do not meet the minimum requirements for high availability. If you need a system that is highly available, see High Availability Installations (HA).
On the initial node, install all the processes except for the backgrounder. Below is the number of instances of the processes for an 8 core machine:

- **VizQL Server:** Set to 2 instances. (default calculation: Number of physical cores divided by 4, up to a maximum of 4).

- **Cache Server, and Data Server:** Set to 2 instances. One instance of Ask Data is automatically configured on the node that has Data Server.

- **Elastic Server:** Elastic Server memory is configured to 1GB by default and can be configured to improve performance by using the `elast-icserver.vmopts` TSM configuration option. For more information, see tsm configuration set Options.

- **All other processes,** only one instance of the process is installed, regardless of hardware. One instance of Interactive Microservice Container is installed on a node that has Application Server enabled, and one instance of Non-Interactive Microservice Container is installed on a node that has Backgrounder enabled.

- **Isolate backgrounder on the additional node.** To calculate the minimum number of backgrounder processes to run on this node, divide the computer’s total number of physical cores by 4. To calculate the maximum number, divide the computer’s total physical cores by 2. In the example shown above, both the nodes are on machines with 8 physical cores. When you install the backgrounder, Tableau Server automatically installs one instance of the Data Engine.

**Note:** This configuration assumes that you do not have Tableau Prep Conductor enabled on your Tableau Server. If are using Tableau Prep Conductor to schedule and manage flows, and have an extract heavy environment, we recommend that you have at least 3 nodes and use the 3 node configuration described later in this topic.

As you monitor and gather data about the performance and usage, you can fine tune and configure the number of instances for these processes. For example, on the node where it is...
dedicated to running backgrounder, initially, you can set the number of backgrounders to minimum (total number of cores divided by four), and increase the number of backgrounder processes later if you find that:

- Extract refreshes are taking a long time to complete
- Subscriptions and alerts are not completing on time

For more information on performance tuning, see Performance Tuning topic.

Two Node Installation - Specialized for flow environments

Start with a two node configuration if you are planning to publish, schedule, and manage flows on your Tableau Server.

**Important:** Two-node configurations do not meet the minimum requirements for high availability. If you need a system that is highly available, see High Availability Installations (HA).
On the initial node, install all the processes. Below is the number of instances of the processes for an 8 core machine:

- VizQL Server: Set to 2 instances. (default calculation: Number of physical cores divided by 4, up to a maximum of 4).

- Cache Server, and Data Server: Set to 2 instances. One instance of Ask Data is automatically configured on the node that has Data Server.

- Backgrounder: Minimum 2, maximum 4. The diagram above shows the maximum for an 8 core node. Tableau Prep Conductor is automatically configured on the node where you have backgrounder installed. On the initial node, set the Backgrounder node role to run all job types including flows using the `tsm topology set-node-role tsm` configuration. For more information, see `tsm topology set-node-role`.

- Elastic Server: Elastic Server memory is configured to 1GB by default and can be configured to improve performance by using the `elast-icserver.vmopts` TSM configuration option. For more information, see `tsm configuration set Options`.

- All other processes, only one instance of the process is installed, regardless of hardware. One instance of Interactive Microservice Container is installed on a node that has Application Server enabled, and one instance of Non-Interactive Microservice Container is installed on a node that has Backgrounder enabled.

- Isolated the backgrounder on the additional node to run only flows. Use the `tsm topology set-node-role tsm` configuration to configure this setting. For more information, see `tsm topology set-node-role`.

Note: If you have both a heavy extract environment, and schedule and manage flows on your server, we recommend that you use the 3 node configuration described below.
High Availability Installations (HA)

Recommendations

A highly available installation of Tableau Server is a distributed installation that is designed to maximize the availability of Tableau Server. High availability basically means that the system is available with minimal amount of downtime. To build in redundancy for HA related items such as repository, file redundancy, and failover, you need a minimum of three nodes. The tolerance for downtime will vary for each organization and depends on the SLAs you have established in your organization.

High availability is achieved by eliminating single points of failure and detecting failures and setting up a reliable failover system. HA in Tableau Server is mainly achieved by:

- File redundancy with multiple File Store/Data Engine instances.
- Active/Passive Repository across two nodes.
- Adding an external load balancer to make sure your installation is robust to Gateway failures and make sure that requests only get routed to functioning Gateway processes.

Server Configuration

Three-node configuration:
To build in redundancy, you need to add additional nodes to host instances of the repository and File Store/Data Engine processes. You can add instances of other processes, including multiple instances of a process on a node.

To build redundancy for the type of backgrounder jobs, have one of the nodes (initial...
node in this example) run all type of jobs. Backgrounders run all types of jobs by default. On one of the additional nodes, set the backgrounder to run only flows, and the other additional node to run all jobs except for flows.

- The successful functioning of Tableau Server depends on a properly functioning Coordination Service. For server installations of three or more nodes, we recommend that you add additional instances of the Coordination Service by deploying a new Coordination Service ensemble. This provides redundancy and improved availability in the event that one instance of the Coordination Service has problems. For more information, see Deploy a Coordination Service Ensemble.

- Elastic Server memory is configured to 1GB by default and can be configured to improve performance by using the `elasticserver.vmopts` TSM configuration option. For more information, see tsm configuration set Options.

- To reduce the system’s vulnerability, you can run multiple gateways and additional instances of some of the server processes. The fewest number of computers required to achieve this configuration is three.

- The repository has also been moved from the initial node to one of the additional nodes, and a second, passive instance has been added to the other new node.

- One instance of Interactive Microservice Container is installed on a node that has Application Server enabled, and one instance of Non-Interactive Microservice Container is installed on a node that has Backgrounder enabled.

**NOTE:** In certain circumstances you may want to limit the processes running on your initial node. Reasons for doing this include wanting to run as few processes as possible on the node to limit processing requests on the node. You might also remove licensed Tableau Server processes from the node if you have a core-based license and do not want the initial node cores to count against your core use. For more information on Tableau Server licensed processes, see Tableau Server Processes from the node.
Virtual Machines and Public Cloud Deployments

In general, the considerations and recommendations described in this topic apply to virtual environment and cloud deployments.

If you are running Tableau Server in a virtual environment, use your VM host’s best practices for vCPU allocation in relation to the number of physical CPU cores on the VM host. Typically 2 vCPUs = 1 physical core for Tableau Server. For example, for AWS installations, the 4 core minimum recommendation is equivalent of 8 AWS vCPUs.

For more information on cloud-based deployments, see:

- Install Tableau Server in the Cloud.

Beyond Baseline Configurations

If you are planning a system whose configuration is beyond the limits documented here, contact Tableau Professional Services.

Disaster Recovery Considerations

While HA configurations reduce downtime, you may still encounter failures in case of a disaster or hardware failures. In addition to the above considerations, you should evaluate the importance of disaster recovery in your organization and plan for a deployment that helps you meet your disaster recovery goals and objectives.

When planning for disaster recovery (DR) in your Tableau environment, there are two main factors to consider:

- Recovery Time Objective (RTO), a measure of how much downtime your business can accept before a full recovery.
  - Influences how often you restore your backups to an alternative cluster and the amount of infrastructure investment.

- Recovery Point Objective (RPO), a measure of how much data loss your business can tolerate.
Influences how often you will need to take backups of your system.

For Tableau Server the RPO cannot be shorter than the time it takes to complete a full backup of your server.

The diagram below illustrates how to plan for a range of RTO requirements:

![Diagram showing the impact of recovery time on RPO](image)

Tableau Server Scalability

These baseline configurations may not be enough as your needs change and grow, and you might need to scale your Tableau Server beyond these configurations. Like other enterprise platforms, Tableau Server scales up by adding processor, memory, and disk to existing node, and scales out by adding more nodes to a cluster. However, scalability and performance are heavily dependent on external systems and user activity. The configuration of your Tableau Server can be different depending on your requirements and variables:
For more information on Tableau Server scalability and the variables affecting scalability, see Tableau Server Scalability whitepaper.

Identity Store

Tableau Server requires an identity store to manage user and group information. There are two kinds of identity stores: local and external. When you install Tableau Server you must configure either a local identity store or an external identity store.

For information about configuration options for the identity store, see identityStore Entity and LDAP Configuration Reference.

Local identity store

When you configure Tableau Server with a local identity store, all user and group information is stored and managed in the Tableau Server repository. In the local identity store scenario, there is no external source for users and groups.

External identity store

When you configure Tableau Server with an external store, all user and group information is stored and managed by an external directory service. Tableau Server must synchronize with the external identity store so that local copies of the users and groups exist in the Tableau Server repository, but the external identity store is the master source for all user and group data.
If you have configured the Tableau Server identity store to communicate with an external LDAP directory, then all users (including the initial admin account) that you add to Tableau Server must have an account in the directory.

When Tableau Server is configured to use an external LDAP directory for authentication, you must first import user identities from the external directory to the identity store. When users sign in to Tableau Server, their credentials are passed to the external directory, which is responsible for authenticating the user; Tableau Server does not perform this authentication. However, the Tableau user names stored in the identity store are associated with rights and permissions for Tableau Server. Therefore, after authentication is verified, Tableau Server manages user access (authorization) for Tableau resources.

Active Directory is an example of an external user store. Tableau Server is optimized to interface with Active Directory. For example, when you install Tableau Server on an Active Directory domain-joined computer using the Configure Initial Node Settings, Setup will detect and configure most Active Directory settings. If, on the other hand, you are using TSM CLI to install Tableau Server, you must specify all the Active Directory settings. In this case, be sure to use the LDAP - Active Directory template to configure identity store.

If you are installing into Active Directory, we recommend that you review User Management in Active Directory Deployments before you deploy.

For all other external stores, Tableau Server supports LDAP as a generic way to communicate the identity store. For example, OpenLDAP is one of several LDAP server implementations with a flexible schema. Tableau Server can be configured to query the OpenLDAP server. To do so, the directory administrator must provide information about the schema. During setup, you must use Configure Initial Node Settings to configure a connection to other LDAP directories.

**LDAP bind**

Clients that wish to query a user store using LDAP must authenticate and establish a session. This is done by binding. There are multiple ways to bind. Simple binding is authenticating with a username and password. For organizations that connect to Tableau Server
with simple bind, we recommend configuring an SSL encrypted connection, otherwise the credentials are sent over the wire in plaintext. Another type of binding Tableau Server supports is GSSAPI binding. GSSAPI uses Kerberos to authenticate. In Tableau Server’s case, Tableau Server is the client and the external user store is the LDAP server.

LDAP with GSSAPI (Kerberos) bind

We recommend binding to LDAP directory with GSSAPI. To bind with GSSAPI you will need a keytab file specifically for the Tableau Server service.

If you are installing into Active Directory, and the computer where you are installing Tableau Server is already joined to the domain, then the computer may already have a configuration file and a keytab file. In this case, the Kerberos files are for the operating system functionality and authentication. Strictly speaking, you can use these files for GSSAPI bind, but we don’t recommend using them. Instead, contact your Active Directory administrator and request a keytab specifically for the Tableau Server service. See Understanding Keytab Requirements.

Assuming your operating system has a properly configured keytab for authentication to the domain, then the Kerberos keyfile for GSSAPI bind is all you need for the base installation of Tableau Server. If you plan to use Kerberos authentication for users, then configure Kerberos for user authentication and Kerberos delegation to data sources after installation is complete.

LDAP over SSL

By default, LDAP with simple bind is not encrypted. If you are configuring LDAP with simple bind, we strongly recommend that you enable LDAP over SSL (LDAPS).

If you already have certificates installed for LDAP on the computer running Tableau Server, then LDAPS should work with minimal configuration during the installation process.

Note: If you are running Tableau Server in a distributed deployment, then you must manually copy the SSL certificate to each node in the cluster. Copy the certificate only to
those nodes where the Tableau Server Application Server process is configured. Unlike other shared files in a cluster environment, the SSL certificate used for LDAP will not be automatically distributed by the Client File Service.

Specifically, if you have installed Tableau Server, and you have valid certificates installed in the Tableau keystore (/etc/opt/tableau/tableau_server-/tableauservicesmanagerca.jks), then you can specify SSL when you configure the identity store.

The password for the Java keystore is changeit. (Do not change the password for the Java keystore).

If you do not have certificates already in place on your computer that are configured for the LDAP server then you must obtain a SSL certificate for the LDAP server and import it into the Tableau system keystore.

Use the "keytool" Java tool to import certificates. In a default installation, this tool is installed with Tableau Server at /opt/tableau/tableau_server-/packages/repository.<installer version>/jre/bin/keytool.

The following command (for RHEL-like distributions) imports the certificate:

```bash
sudo "$PROGRAMFOLDER"/packages/repository*/jre/bin/keytool -import -file "$CERTSDIR" -alias "$OPENLDAPSSLSERVER" -keystore /etc/opt/tableau/tableau_server/tableauservicesmanagerca.jks -storepass changeit -noprompt
```

System user and groups

Tableau Server on Linux uses one user, and two groups for proper operation. The user and groups can be local or from an LDAP directory service.
Tableau Server on Linux Administrator Guide

User

Tableau Server requires a service account. This account is an unprivileged user with normal login privileges.

LDAP/AD

If you want to use an existing user account in your LDAP directory, or if you want to create a new user account in your LDAP directory for Tableau Server then you must disable account and group creation during installation.

Specifically, you will need to set the `--disable-account-creation` and `-g` flags when you run the initialize-tsm script. If you are not going to use the default names, then you will also need to specify user and group names with the `--unprivileged-user` and the `--tsm-authorized-group` flags. See Help Output for initialize-tsm Script for more details.

The user account should be an unprivileged user with normal login privileges. Configure the account with the following characteristics:

- Shell set to `/bin/bash`.
- For convenience, consider setting the home directory to the data directory path. The account must have ownership and write privileges to the home directory.

Local

If you want to use local users and groups, the `initialize-tsm` script can automatically create them during installation. By default, the unprivileged account is named `tableau`. If an account named `tableau` already exists, the Tableau will use it. To specify a different account, see Help Output for initialize-tsm Script.

Groups

Tableau Server requires two groups for operation.

In a default installation, the local `tableau` account belongs to a primary group named `tableau`. However, if you specify an alternate unprivileged user during installation, then the
primary group for that alternate account will be used. As a convenience, any account can be added to this group to be able to read the Tableau Server log files (without becoming root).

The second group is used to authorize which users are authorized to authenticate to Tableau Services Manager (TSM). Any user in this group will be able to send commands to TSM, so it should be restricted to Tableau Server administrators. By default, this group is named `tsadmin`.

If there is an existing LDAP group that matches these criteria, it can be used. Otherwise, a new LDAP group could be created, and all Tableau Server administrators can be added to this new group. Note that at least one administrator must be in this group in order to successfully initialize Tableau Server.

**Authenticating clients**

Basic user authentication in Tableau Server is by username and password sign-in for both local and external user stores. In the local case, user passwords are stored as a hashed password in the repository. In the external case, Tableau Server passes the credentials to the external user store and awaits a response as to whether the credentials are valid. External user stores can also handle other kinds of authentication like Kerberos or SSPI (Active Directory only), but the concept is still the same, Tableau Server delegates the credentials or user to the external store and awaits a response.

You can configure Tableau Server such that username-password sign-in is disabled. In these scenarios other authentication methods, such as trusted authentication, OpenID, or SAML can be used. See Authentication.

**User Management in Active Directory Deployments**

This topic describes important technical details that you should be familiar with if you use Active Directory to authenticate users for Tableau Server. This topic assumes that you are familiar with Active Directory user management and basic Active Directory schema and domain concepts.
Note: If you run Tableau Server on Linux, all external directory communication is configured and managed with a LDAP identity store. In the context of user and group synchronization, Tableau Server configured with LDAP identity store is equivalent to Active Directory. Active Directory synchronization features in Tableau Server function seamlessly with properly configured LDAP directory solutions.

Active Directory user authentication and Tableau Server

Tableau Server stores all user names in the Tableau Server identity store, which is managed by the repository. If Tableau Server is configured to use Active Directory for authentication, you must first import user identities from Active Directory to the identity store. When users sign in to Tableau Server, their credentials are passed to Active Directory, which is responsible for authenticating the user; Tableau Server does not perform this authentication. (By default, NTLM is used for authentication, but you can enable Kerberos or SAML for single sign-on functionality—however, in all these cases, authentication is left to Active Directory.) However, the Tableau user names stored in the identity store are associated with rights and permissions for Tableau Server. Therefore, after authentication is verified, Tableau Server manages user access (authorization) for Tableau resources.

Active Directory user name attributes and Tableau Server

Active Directory uniquely identifies user objects using several attributes. (For details, see User Naming Attributes on the MSDN website.) Tableau Server relies on two Active Directory user naming attributes:

- **sAMAccountName.** This attribute specifies the logon name that was originally designed for use with older versions of Windows In many organizations, this name is combined with the NetBIOS name for authentication, using a format like `example\j-smith`, where `example` is the NetBIOS name and `j-smith` is the `sAMAccountName` value. Due to the original design in Windows, the `sAMAccountName` value must be less than 20 characters.
In the Windows Active Directory Users and Computers administrative console, this value is in the field labeled User logon name (pre-Windows 2000) on the Account tab of the user object.

- userPrincipalName (UPN). This attribute specifies a user name in the format jsmith@example.com, where jsmith is the UPN prefix and @example.com is the UPN suffix.

In the Windows Active Directory Users and Computers administrative console, the UPN is a concatenation of two fields on the Account tab of the user object: the User logon name field, and the domain drop-down list next to it.

Adding users from Active Directory

You can add users individually from Active Directory, either by typing them in the server environment or by creating a CSV file and importing the users. You can also add Active Directory users by creating a group via Active Directory and importing all of the group's users. The result can be different depending on which approach you’re using.

Adding user groups

If you import an Active Directory user group, Tableau will import all users from the group using the sAMAccountName.

Sync behavior when removing users from Active Directory

Users cannot be automatically removed from Tableau Server through an Active Directory sync operation. Users that are disabled, deleted, or removed from groups in Active Directory remain on Tableau Server so that you can audit and reassign the user’s content before removing the user’s account completely.

However, Tableau Server will act upon user objects differently based how the status of that user object changes in Active Directory. There are two scenarios: deleting/disabling users in Active Directory or removing users from synchronized groups in Active Directory.
Tableau Server on Linux Administrator Guide

When you delete or disable a user in Active Directory and then synchronize that user’s group on Tableau Server, the following occurs:

- The user is removed from the Tableau Server group you synchronized.
- The user’s role is set to "unlicensed."
- The user will still belong to the All Users group.
- The user is unable to sign in to Tableau Server.

When you remove a user from a group in Active Directory and then synchronize that group on Tableau Server, the following occurs:

- The user is removed from the Tableau Server group you synchronized.
- The users role is retained: it is not set to "unlicensed."
- The user will still belong to the All Users group.
- The user will still have permission to the Tableau Server with access to everything that the All Users group is granted permission to use.

In both instances, to remove a user from Tableau Server, the server administrator must delete the user from the Server Users page in Tableau Server.

Domain nicknames

In Tableau Server, domain nickname is equivalent to the Windows NetBIOS domain name. In a Windows Active Directory forest, a fully qualified domain name (FQDN) can have an arbitrary NetBIOS name. The NetBIOS name is used as the domain identifier when a user logs in to Active Directory.

For example, the FQDN west.na.corp.lan might be configured with a NetBIOS name (nickname) of SEATTLE. The user jsmith in that domain could log on to Windows using either of the following user names:

- west.na.corp.example.com\jsmith
- SEATTLE\jsmith

If you want your users to sign in to Tableau Server with a NetBIOS name instead of the FQDN, then you'll need to verify that the nickname value for each domain where users log in is set. See editdomain for information on how to view and set the nickname value for each domain.
Support for multiple domains

You can add users from a domain that's different from the domain of the Tableau Server computer in these cases:

- Two-way trust has been established between the server’s domain and the users’ domain.

- The server’s domain trusts the users’ domain (one-way trust). See Domain Trust Requirements.

The first time you add a user from the non-server domain, use the fully-qualified domain name with the user name. Any additional users you add from that domain can be added using the domain’s nickname, provided the nickname matches the NetBIOS name.

Duplicate display names

If user display names are not unique across multiple domains, then managing users with the same display name in Tableau can be confusing. Tableau Server will display the same name for two users. For example, consider an organization with two domains, example.lan and example2.lan. If user John Smith exists in both domains, then adding that user to groups and other administrative tasks will be confusing in Tableau Server. In this scenario, consider updating the display name in Active Directory for one of the users to differentiate the accounts.

Sign in to Tableau Server with NetBIOS name

Users can sign in to Tableau Server using the domain nickname (NetBIOS name), for example, SEATTLE\jsmith.

Tableau Server cannot query for NetBIOS name for a given FQDN. As a result, Tableau sets the nickname of a given FQDN according to the first entry in the namespace. For example, given the FQDN west.na.corp.lan, Tableau sets the nickname to west.

Therefore, you might need to update the domain nickname on Tableau Server before users can sign in using the nickname. If you do not update the nickname, users will have to sign in
using a fully qualified domain name. For more information, see Users From New Domain Unable to Log In and Do Not Appear in User List in the Tableau Knowledge Base.

Domain Trust Requirements

When you run Tableau Server in an Active Directory environment across multiple domains (either in the same Active Directory forest or in different forests), some Tableau functionality is dependent on the trust relationship between the domains. For example, some administrators manage users in domains that are separate from where they deploy server applications, such as Tableau Server. In other organizations, a Tableau Server deployment might be shared with external partners or with different partners in the organization. Finally, Windows-authenticated data sources, such as SQL Server, MSAS, or Oracle, that Tableau Server connects to may also be in other domains.

If it’s feasible, we recommend configuring two-way trust between all domains that interact with Tableau Server. If this is not possible, Tableau Server can be configured to support user authentication where a one-way trust has been configured. In this case, a one-way trust between domains is supported when the domain in which Tableau Server is installed is configured to trust the domain where user accounts reside.

The following illustration shows one-way trust between the domain where Tableau Server is installed and the domain where user accounts reside:
In this scenario, Tableau Server is in the dev.local domain, and users from the users.lan Active Directory domain are imported into Tableau Server. A one-way trust is required for this scenario; specifically, the dev.local domain is configured to trust the users.lan domain. Users in the users.lan domain can access Tableau Server in the dev.local with their normal Active Directory credentials. However, you may need to update the domain nickname on Tableau Server before users log on with the nickname. Refer to the Tableau Knowledge Base for more information.

When you configure Tableau Server for this scenario, specify the user domain during installation. See Configure Initial Node Settings.

Kerberos single sign-on is supported in this one-way trust scenario.

Review User Management in Active Directory Deployments to understand how multiple domains, domain naming, NetBIOS, and Active Directory user name format influence Tableau user management.

Connecting to live data in one-way trust scenarios

In the one-way trust scenario, users connecting to Tableau Server can connect to live data that's hosted in the cloud or on any other data source on premises that does not rely on Windows authentication.

Data sources that require Windows-authentication might have additional authentication requirements that complicate the scenario, or that can even prevent Tableau Server users from connecting. This is because Tableau Server uses a service account for authentication with such data sources. If you are running Tableau Server in a different domain than data sources that use Windows authentication, verify that the service account that is used for Tableau Server can access the data source.

Communicating with the Internet

In most enterprises, Tableau Server needs to communicate with the internet. Tableau Server was designed to operate inside a protected internal network. Do not set up Tableau Server directly on the internet or in a DMZ. Instead, communications between your network
Tableau Server on Linux Administrator Guide

and the internet should be mediated using proxy servers. If the computer running Tableau Server cannot access the internet directly, then you may need to deploy forward proxy servers to mediate traffic from inside the network to targets on the internet. Tableau Server doesn’t support pass-through or manual proxy authentication.

For inbound traffic, we recommend running Tableau Server behind reverse proxy servers.

How Tableau communicates with the internet

Tableau Server requires outbound access to the internet for these scenarios:

- Working with maps. Tableau uses map data that is hosted externally.

  Tableau Server needs to connect to the following internet locations with port 443 to use maps:
  
  - mapsconfig.tableau.com
  - api.mapbox.com

  If Tableau cannot make these connections, maps may fail to load.

  You can test connectivity by accessing each of those addresses in a browser: https://mapsconfig.tableau.com/v1/config.json and https://api.mapbox.com/ will prompt you to download a json file.

  For Tableau Server version 2019.1 and earlier, see the documentation for your version: Tableau Help

- Connecting to the Tableau send-logs server.

  You can upload log files to Tableau when working with Support. See tsm maintenance send-logs. To successfully upload files to Tableau, your Tableau Server must be able to communicate with the send-logs server on port 443:

  report-issue.tableau.com:443

- Licensing. Tableau products connect to the internet to activate product keys. Unless
you activate Tableau software with the **Offline Activation Tool**, all Tableau products must have access to the internet to validate licenses. Specifically Tableau requires internet access during the following licensing operations: activation, deactivation, and on the refresh maintenance date. For more information about these operations, see **Manage Licenses**.

Tableau Server needs to connection to the following internet locations for licensing purposes:

- licensing.tableau.com:443
- atr.licensing.tableau.com:443
- s.ss2.us
- ocsp.rootg2.amazontrust.com
- ocsp.rootca1.amazontrust.com
- ocsp.sca1b.amazontrust.com
- crt.sca1b.amazontrust.com
- crt.rootca1.amazontrust.com
- ocsp.sca0a.amazontrust.com
- crt.sca0a.amazontrust.com
- ocsp.sca1a.amazontrust.com
- crt.sca1a.amazontrust.com
- ocsp.sca2a.amazontrust.com
- crt.sca2a.amazontrust.com
- ocsp.sca3a.amazontrust.com
Requests to the above domains may be on port 80 or 443.

If Tableau Server cannot make a connection while attempting to activate its license, you will be prompted to do an offline activation.

- Working with external or cloud-based data.

Tableau Server needs to connect to the following internet location for Box, Dropbox, and OneDrive services:

```
galop.connectors.tableau.com:443
```

Tableau Server can run without internet access. For more information about deploying Tableau Server in organizations without access to the internet, see Install Tableau Server in a Disconnected (Air-Gapped) Environment.

In many enterprises, users also need to access Tableau Server from outside the network (that is, from the internet). For example, in many enterprises, users want to be able to reach Tableau Server from their mobile devices in order to interact with views that are stored on the server. To configure access to Tableau Server from the internet or from mobile devices, you should use a reverse proxy. See Configuring Proxies for Tableau Server.
Configuring Proxies for Tableau Server

In most enterprises, Tableau Server needs to communicate with the internet. Tableau Server was designed to operate inside a protected internal network. Do not set up Tableau Server directly on the internet or in a DMZ. Instead, communications between your network and the internet should be mediated using proxy servers. Forward proxy servers mediate traffic from inside the network to targets on the internet. Reverse proxy servers mediate traffic from the internet to targets inside the network.

Who should read this article?

This article is for IT professionals who are experienced with general networking and gateway proxy solutions. The article describes how and when Tableau requires internet access, and describes how to configure your network and Tableau to use forward and reverse proxy servers for access to and from the internet. There are many third-party proxy solutions available, so some of the content in the article is necessarily generic.

Before you configure a proxy server, see Communicating with the Internet.

Configure a forward proxy server

To enable communication from Tableau Server to the internet, deploy Tableau Server behind a forward proxy server. When Tableau Server needs access to the internet, it doesn't send the request directly to the internet. Instead, it sends the request to the forward proxy, which in turn forwards the request. Forward proxies help administrators manage traffic out to the internet for tasks such as load balancing, blocking access to sites, etc.

If you use a forward proxy, you must configure the computers that run Tableau Server inside the network to send traffic to the forward proxy. Tableau Server doesn't support pass-through or manual proxy authentication.

Configuring Tableau Server on Linux to work with a forward proxy

We recommend configuring Tableau Server to work with your forward proxy solution as part of the installation process. Specifically, configure Tableau Server when you run
Tableau Server on Linux Administrator Guide

./initialize-tsm as described in Install and Initialize TSM, or as part of Automated Installation of Tableau Server.

The procedure below describes how to create a forward proxy configuration file for Tableau Server on Linux.

The configuration file is stored in the following directory:

~<unprivileged_user>/.config/systemd/tableau_server.conf.d

By default, Tableau Server creates the unprivileged user, tableau. Therefore, the default path to the configuration directory is:

~tableau/.config/systemd/tableau_server.conf.d

The proxy configuration file name in this topic and in the configuration file below is referred to as 20-proxy.conf. You can name this file according to your own convention, but it must use the .conf extension. systemd will process files stored in the tableau_server.conf.d directory in lexical order according to file name.

1. Run the tsm stop command.

2. Start a session as the unprivileged user. By default, tableau, is the unprivileged user created by Tableau Server during installation. Run the following command:

   sudo su -l tableau

3. Create or open the 20-proxy.conf file in the tableau_server.conf.d directory. If you configured forward proxy during setup, then the 20-proxy.conf file has already been created.

   • Create the file. Run the following command:

     touch ~tableau/.config/systemd/tableau_server.conf.d/20-proxy.conf
Open the 20-proxy.conf file in a text editor.

4. Copy the Proxy configuration file contents into the file. If you are editing an existing file, take care not to delete the configuration. The Proxy configuration file contents include instructions on how to set forward proxy configurations. After you have edited and saved the file go to Step 5.

Proxy configuration file contents

# Always edit this file on Tableau Server as the unprivileged user. By default, tableau, is the unprivileged user created by Tableau Server during installation.
# Set environment variables http_proxy and https_proxy to point to your proxy host.
# For example, to set the proxy to example-host for ports 80 and 443, run the following commands:
#
http_proxy=http://example-host:80/
https_proxy=http://example-host:443/
#
# Take care to use 'http' when you specify the URL for the https_proxy environmental variable.
# Do not specify the 'https' protocol for the value of the https_proxy environmental variable.
#
# To bypass the proxy server, specify exceptions in the no_proxy variable. Use this variable if your proxy server does not route internal addresses.
# You must also add exceptions to this proxy configuration to guarantee that all communications within a local
Tableau Server on Linux Administrator Guide

Tableau Server cluster (if you have one now or will have one later) do not route to the proxy server.
# Enter both the host name and the IP address for each computer. Additionally, include the canonical host name (localhost) and IP address (127.0.0.1) for the local computer.
# For example, to specify exceptions for a three-node cluster:

#
# no_proxy=
"localhost,127.0.0.1,hostname1,hostname2,hostname3,IP1,IP2,IP3"
#
# Where "hostname1" is the actual hostname of node 1, and "IP1" is the actual IP address of node 1, etc.

5. Exit the Tableau shell. Run the following command:

exit

6. Restart the TSM business services. Run the following script:

    sudo /opt/tableau/tableau_server-
    /packages/scripts.<version>/start-administrative-services

7. Restart TSM.

    tsm restart

How a reverse proxy works with Tableau Server

A reverse proxy is a server that receives requests from external (internet) clients and forwards them to Tableau Server. Why use a reverse proxy? The basic answer is security. A reverse proxy makes Tableau Server available to the internet without having to expose the individual IP address of that particular Tableau Server to the internet. A reverse proxy also acts as an authentication and pass-through device, so that no data is stored where people
outside the company can get to it. This requirement can be important for organizations that are subject to various privacy regulations such as PCI, HIPAA, or SOX.

The following diagram illustrates the communication path when a client makes a request to Tableau Server that is configured to work with a reverse proxy server.

1. An external client initiates a connection to Tableau Server. The client uses the public URL that's been configured for the reverse proxy server, such as https://tableau.example.com. (The client doesn't know that it's accessing a reverse proxy.)

2. The reverse proxy maps that request in turn to a request to Tableau Server. The reverse proxy can be configured to authenticate the client (using SSL/TLS) as a pre-condition to passing the request to Tableau Server.

3. Tableau Server gets the request and sends its response to the reverse proxy.

4. The reverse proxy sends the content back to the client. As far as the client is concerned, it just had an interaction with Tableau Server, and has no way to know that the communication was mediated by the reverse proxy.

Proxy servers and SSL

For better security, you should configure reverse proxy servers to use SSL for any traffic that's external to your network. This helps to ensure privacy, content integrity, and authentication. Unless you’ve deployed other security measures to protect traffic between your internet gateway and Tableau Server, we also recommend configuring SSL between the
gateway proxy and Tableau Server. You can use internal or self-signed certificates to encrypt traffic between Tableau Servers and other internal computers.

Mobile access

Tableau Server adds an X-header to all HTTP responses for Tableau Mobile sessions. By default, most proxy solutions will preserve X-headers. If your proxy solution does not preserve X-headers, then you will need to configure your proxy server to preserve the following header to all HTTP responses for Mobile client sessions: X-Tableau: Tableau Server.

If you have configured authentication at the proxy server gateway, then your proxy server must respond to Tableau Mobile HTTP requests with a HTTP 302 response. The 302 must include a redirect to the identity provider login page. To view a diagram that describes the 302 authentication sequence, see Tableau Mobile Authentication Sequence in the Tableau Community.

Reverse proxy and user authentication

Tableau Server will always authenticate users. This means that even if you are authenticating inbound connections at the gateway for your organization, Tableau Server will still authenticate the user.

However, not all clients will support user authentication with a reverse proxy:

- For supported web browsers, you can use SAML, OpenID Connect, Kerberos, Trusted Tickets or manual authentication with a reverse proxy. However, we recommend a transparent scenario where user requests are not prompted for authentication at the gateway. This recommendation doesn't prohibit using SSL for client/server system-level authentication at the gateway proxy, in fact, we strongly recommend SSL system-level authentication.

- Tableau Mobile supports SAML or manual authentication with a reverse proxy. The iOS version of Tableau Mobile additionally supports Kerberos with a reverse proxy.
The same recommendation above applies.

- Tableau Desktop does not support authentication with a reverse proxy. To support remote access with Tableau Desktop, use a VPN solution or configure your reverse proxy to route traffic from Tableau Desktop directly to Tableau Server for authentication.

If your organization is authenticating with Active Directory:

- Active Directory with Enable automatic logon (SSPI) is not supported with a reverse proxy.
- Tableau Server must be configured for reverse proxy before configuring Tableau Server for Kerberos. For more information, see Configure Kerberos.

Configure Tableau Server to work with a reverse proxy server

Before you configure Tableau Server, you'll need to collect the following information about the proxy server configuration. To configure Tableau Server, you use the `tsm configuration set` command. The information you need to collect corresponds to options you'll need when you run `tsm`.

Most of the following `tsm` options are also used to configure Tableau Server deployments that operate behind a load balancer. For more information, see Add a Load Balancer.

<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
<th>Corresponding <code>tsm configuration set</code> option</th>
</tr>
</thead>
</table>
| IP address or CNAME | You can either enter an IP address or a CNAME for this option.  
The public IP address or addresses of the proxy server. The IP address must be in IPv4 format, such as `203.0.113.0`, and it must be a static IP. | `gateway.trusted` |
<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
<th>Corresponding tsm configuration set option</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>If you are unable to provide a static IP, or if you are using cloud proxies or external load balancers, you can specify the CNAME (Canonical Name) DNS value that clients will use to connect to Tableau Server. This CNAME value must be configured on your reverse proxy solution to communicate with Tableau Server.</td>
<td></td>
</tr>
<tr>
<td>FQDN</td>
<td>The fully qualified domain name that people use to reach Tableau Server, such as tableau.example.com. Tableau Server doesn't support context switching for this option. For example, the following URL is not supported: example.com/tableau.</td>
<td>gateway.public.host</td>
</tr>
<tr>
<td>Non-FQDN</td>
<td>Any subdomain names for the proxy server. In the example of tableau.example.com, the subdomain name is tableau.</td>
<td>gateway.trusted_hosts</td>
</tr>
<tr>
<td>Aliases</td>
<td>Any public alternative names for the proxy server. In most cases, aliases are designated using CNAME values. An example would be a proxy server bigbox.example.com and CNAME entries of ftp.example.com and <a href="http://www.example.com">www.example.com</a> and</td>
<td>gateway.trusted_hosts</td>
</tr>
</tbody>
</table>
If you are using a distributed installation of Tableau Server, then run the following `tsm` commands on the initial node in your cluster.

1. Enter the following command to set the FQDN that clients will use to reach Tableau Server through the proxy server, where `name` is the FQDN:

   ```
   tsm configuration set -k gateway.public.host -v 'name'
   ```

   For example, if Tableau Server is reached by entering `https://tableau.example.com` in the browser, enter this command:

   ```
   tsm configuration set -k gateway.public.host -v 'tableau.example.com'
   ```

2. Enter the following command to set the address or the CNAME of the proxy server, where `server_address` is the IPv4 address or CNAME value:

   ```
   tsm configuration set -k gateway.trusted -v 'server_ip_address'
   ```

   If your organization uses multiple proxy servers, enter multiple IPv4 addresses, separating them with commas. IP ranges are not supported. To improve start up and initialization of Tableau Server, minimize the number of entries for `gateway.trusted`.

3. Enter the following command to specify alternate names for the proxy server, such as its fully qualified domain name, any not fully qualified domain names, and any aliases. If there's more than one name, separate the names with a comma.
Tableau Server on Linux Administrator Guide

tsm configuration set -k gateway.trusted_hosts -v 'name1, name2, name3'

For example:

tsm configuration set -k gateway.trusted_hosts -v
'proxy1.example.com, proxy1, ftp.example.com, www.example.com'

4. If the proxy server is using SSL to communicate with the internet, run the following command, which tells Tableau that the reverse proxy server is using port 443 instead of port 80:

```
tsm configuration set -k gateway.public.port -v 443
```

**Note:** If the proxy server is using SSL to communicate with Tableau Server, SSL must be configured and enabled on Tableau Server.

5. Enter the following command to commit the configuration change:

```
tsm pending-changes apply
```

The `pending-changes apply` command displays a prompt to let you know this will restart Tableau Server if the server is running. The prompt displays even if the server is stopped, but in that case there is no restart. You can suppress the prompt using the `--ignore-prompt` option, but this does not change the restart behavior. For more information, see `tsm pending-changes apply`.

Configure the reverse proxy server to work with Tableau Server

When a client accesses Tableau Server through a reverse proxy, specific message headers have to be preserved (or added). Specifically, all proxy servers in the message chain must be represented in the `gateway.trusted` and `gateway.trusted_hosts` settings.
The following graphic shows example headers for a single-hop message chain, where the proxy server is communicating directly with Tableau Server:

![Single-hop message chain diagram]

The following graphic shows example headers for a multiple-hop message chain, where the message traverses two proxy servers before connecting to Tableau Server:

![Multiple-hop message chain diagram]

The following table describes what these headers are and how they relate to the configuration settings on Tableau Server:

<table>
<thead>
<tr>
<th>Headers</th>
<th>Description</th>
<th>Related Tableau Server settings</th>
</tr>
</thead>
<tbody>
<tr>
<td>REMOTE_ADDR and X-FORWARDED-FOR (XFF)</td>
<td>Tableau Server needs these headers to determine the IP address of origin for requests. X-FORWARDED-FOR header must present IP address chain to Tableau Server in the order the connections have occurred.</td>
<td>The IP address that you set in gateway.trusted must match the IP presented in REMOTE_ADDR. if you sent multiple addresses in gateway.trusted, one of them must match the IP presented in REMOTE_ADDR.</td>
</tr>
</tbody>
</table>
### ADDR.

<table>
<thead>
<tr>
<th>HOST and X-FORWARDED HOST (XFH)</th>
<th>These headers are used to generate absolute links to Tableau Server when it replies to the client. X-FORWARDED-HOST header must present host names to Tableau Server in the order the connections have occurred.</th>
<th>The host names that are presented in X-FORWARDED-HOST header must be included in the host names that you specify in gateway.trusted_hosts.</th>
</tr>
</thead>
<tbody>
<tr>
<td>X-FORWARDED-PROTO (XFP)</td>
<td>This header is required if SSL is enabled for traffic from the client to the proxy, but not for traffic from the proxy to Tableau Server. The X-FORWARDED-PROTO headers are important for scenarios where HTTP or HTTPS is not maintained along each hop of the message route. For example, if the reverse proxy requires SSL for outside requests, but traffic between the reverse proxy and Tableau Server is not configured to use SSL, X-FORWARDED-PROTO headers are required. Some proxy solutions add the X-FORWARDED-PROTO headers automatically, while others do not. Finally, depending on your proxy solution, you might have to configure port forwarding to translate the request from port 443 to port 80.</td>
<td>Port configuration on reverse proxy (inbound connections from client and outbound connections to Tableau Server) must be specified in the corresponding parameter: gateway.public.port, which is the port clients use to connect to the proxy. If the proxy server is using SSL to communicate with Tableau Server, SSL must be configured and enabled on Tableau Server.</td>
</tr>
</tbody>
</table>
Validate reverse proxy setup

To validate your reverse proxy setup, publish workbooks and datasources using Tableau Server web authoring or Tableau Desktop. If you are connecting with a web browser to Tableau Server from the internet, verify that you are using a recommended browser. Publish and view workbooks that use existing datasources as well as a datasource that you’ve published. Use the links below to familiarize yourself with connecting with Tableau Server as an end-user.

<table>
<thead>
<tr>
<th>Task</th>
<th>Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Overview of web authoring.</td>
<td>Use Tableau on the Web</td>
</tr>
<tr>
<td>Log in to Tableau Server from Tableau Desktop or a web browser.</td>
<td>Sign in to Tableau Server or Online</td>
</tr>
<tr>
<td>Publish a workbook to Tableau Server.</td>
<td>Publish a Workbook</td>
</tr>
<tr>
<td>Publish a data source.</td>
<td>Publish a Data Source</td>
</tr>
<tr>
<td>Open workbook from Tableau Server.</td>
<td>Opening Workbooks from the Server</td>
</tr>
<tr>
<td>Log out Server (with Desktop).</td>
<td>Sign in to Tableau Server or Online</td>
</tr>
<tr>
<td>Download workbook from a web browser.</td>
<td>Download Workbooks</td>
</tr>
<tr>
<td>Check to make sure tabcmd (from a non-server client) works.</td>
<td>tabcmd</td>
</tr>
</tbody>
</table>
Tableau Server on Linux Administrator Guide

Related topics

- Tableau Desktop Internet Access Requirements
- Tableau Community: Cannot Create Firewall Rules to Access Mapping Servers
- Add a Load Balancer
Deploy

The topics in this section provide information on installing, configuring, and upgrading Tableau Server on Linux.

Looking for Tableau Server on Windows? See Install and Configure Tableau Server

Install and Configure Tableau Server ................................................................. 73
Automated Installation of Tableau Server ......................................................... 135
Install Tableau Server in a Disconnected (Air-Gapped) Environment .............. 142
Install Tableau Server in the Cloud ................................................................. 150
Post Installation Tasks ...................................................................................... 151
Distributed and High Availability Tableau Server Installations ....................... 189
Migrate ............................................................................................................... 302
Upgrade Tableau Server Overview ................................................................. 341
Uninstall Tableau Server .................................................................................. 430

Install and Configure Tableau Server

The topics referenced at the bottom of this page describe the steps to install and configure Tableau Server. If you are installing a distributed deployment (cluster), use the steps in this topic to install the initial node, then, to install additional nodes, see Distributed and High Availability Tableau Server Installations.

After you run the installation, you must then continue setup by activating a license, registering Tableau Server, and configuring various settings including authentication.
Other installation methods

There are a few alternative methods that you can use to install Tableau Server.

- If you want a quick start procedure to install Tableau Server in a non-production environment, see Jump-start Installation.

- Automated Installation of Tableau Server.

- If you are installing Tableau Server in an environment without a connection to the internet, see Install Tableau Server in a Disconnected (Air-Gapped) Environment.

- You can also install Tableau Server onto various cloud platforms. See Install Tableau Server in the Cloud.

- For single-server installations in departments or small businesses, see Everybody's Install Guide.

Before you begin

To install Tableau Server you must have a computer that satisfies the hardware requirements. You will get an informational message if your computer meets the minimum requirements but does not satisfy the recommended minimum requirements. In this case, your computer hardware can handle a trial installation of Tableau but is not adequate for a production environment. For more information, see Before you install....

Installation steps

The following steps describe how to install Tableau Server on a single computer. Use the steps to install Tableau Server in a single server deployment. Use the steps to install the initial node in a multi-node Tableau Server deployment. Run the steps sequentially.

1. Install and Initialize TSM

2. Activate and Register Tableau Server
3. Configure Initial Node Settings

4. Add an Administrator Account

5. Validate Installation

Before you install...

**Note:** You can find additional information about technical specifications for Tableau Server on the Tableau web site, [here](#).

This topic includes requirements and recommendations that you must consider before you install Tableau Server into a production environment.

If you want to install a single server, or if you want to do a minimal installation for test purposes, refer to our single-server installation guide, *Everybody's Install Guide*.

If you are deploying Tableau Server in a distributed cluster, review Distributed Requirements in addition to the requirements and recommendations described in this topic.

If you are migrating from Tableau Server on Windows to Tableau Server on Linux, see Migrate Tableau Server from Windows to Linux.

**Hardware recommendations**

The following list describes the minimum hardware recommendations for a single-node installation of Tableau Server:

- 8 core, 2.0 GHz or higher processor
- 64-bit processor architecture
- 32 GB memory
• 50 GB disk space available, with a minimum of 15 GB allocated to the /opt directory, and the remainder allocated to the /var directory for data storage. See the section, Data directory, below.

**Important:** The disk space requirement cannot be checked until you initialize TSM. If you don’t have enough space, you won’t be told this until after you install the Tableau Server package.

To see the full list of recommendations and to see the minimum requirements, see Minimum Hardware Requirements and Recommendations for Tableau Server.

For public cloud deployments on Amazon Web Services and Google Cloud Platform, their “vCPU” is actually a CPU hyper-thread, and not a full CPU core. When sizing cloud instances, you will need twice as many vCPU as the Tableau Server CPU core requirements given (4 vCPU for a minimum trial installation, 16 vCPU for a recommended installation).

**Operating system requirements**

The following distributions of Linux are supported:

• Red Hat Enterprise Linux (RHEL) 7.3 and higher, CentOS 7.3 and higher, Oracle Linux 7.3 and higher, and Amazon Linux 2.

These distributions are collectively referred to in this documentation as RHEL-like.

RHEL 8 is not supported.

• The latest versions of Ubuntu 16.04 LTS and 18.04 LTS.

Non-LTS releases are not supported. Ubuntu version 17.04 is not supported.

• Debian 9

Previous versions of CentOS and Ubuntu are not supported because Tableau Server requires systemd for process management. The version of the installer with the file suffix, .deb, installs on both Ubuntu and Debian distributions.
Custom kernels are not supported.

In a multi-node installation of Tableau Server, all of the computer nodes where you are installing Tableau must run Linux and the same distribution of Linux.

**Note:** All nodes in a multi-node cluster must have the same type of operating system. You cannot install a multi-node instance of Tableau Server on a combination of Linux and Windows nodes.

### Installation directory

The core Tableau Server bits are installed in the `/opt` directory by default. You can change the install directory on RHEL-like systems, but cannot change this on Ubuntu. You cannot specify a symbolic link or a directory location on a Network File System (NFS) volume. The directory where you install Tableau Server must have at least 15 GB of free disk space. If you attempt to install Tableau Server on a computer that does not have enough space, the Tableau Server package will install, but you will be unable to continue with setup.

### Data directory

By default, Tableau Server will create a data directory for all content and extracts that are managed by Tableau. The directory is created at `/var/opt/tableau/tableau_server`.

You can specify a different directory for data (extract) storage during installation. If you plan to use a different directory, do not create the directory. Instead, let Tableau Server setup create the directory. The data directory requires specific permissions that are set during the installation process.

To change the data directory, you must pass a flag along with the data directory path when you run the `initialize/tsm` script. See Help Output for `initialize/tsm` Script.

If you are changing the default data directory:
Do not specify a symbolic link or a data directory location on a Network File System (NFS) volume.

Do not specify a data directory location with a path that includes a period or space. If there is a period or space in the path, initialization will fail.

The data directory must be installed into a different directory than the installation directory.

Important: You cannot change the data directory location after you’ve run initialize-tsm. The data directory location will persist for the life of the deployment, including subsequent upgrades.

Tableau Prep Conductor

Tableau Prep Conductor is one of the processes on Tableau Server. It runs a flow, checks connection credentials, and sends alerts if a flow fails. Tableau Prep Conductor leverages the scheduling and tracking functionality of Tableau Server so you can automate running flows to update the flow output instead of logging into Tableau Prep Builder to manually run individual flows as your data changes.

Tableau Prep Conductor is licensed separately and is available through the Data Management Add-on license. For more information on Tableau Prep Conductor licensing, see Licensing Tableau Prep Conductor.

We recommend you enable Tableau Prep Conductor on a dedicated node. For more information:

- If you are installing a new Tableau Server, see Installing Tableau Server with Prep Conductor.

- If you are adding Tableau Prep Conductor to an existing installation of Tableau Server, see Add Tableau Prep Conductor to your Tableau Server Installation.

Additional requirements

Make sure that your environment also meets the following additional requirements:
Hostname

- Tableau Server must be able to resolve the hostname to an IP address either using the domain name server (DNS) or with a local host file on the computer running Tableau Server. By default, host files are stored at /etc/hosts.
- The hostname of the server must not change after you start Tableau Services Manager during the setup process. For example, this might happen if you use the cloud-init package to initialize a virtual machine, and you install Tableau Server on that virtual machine.
- Hostnames that include underscores (_) are not supported by Tableau Server on Linux.

Static IP address

The computer where you install Tableau Server must have a static IPv4 or IPv6 address.

Database drivers

To connect to specific data sources, the computer where you install Tableau Server must have the correct database drivers installed. For more information, see Database Drivers.

Available ports

TSM and Tableau Server each require an available TCP port in order for you to access them. TSM defaults to port 8850, and the Tableau Server Gateway service defaults to port 80. We strongly recommend that you ensure that both port 8850 and 80 are not in use on your system before installing Tableau Server. If those ports are not available, the TSM and gateway ports may be dynamically remapped to different port numbers, and there is currently no interface for displaying which port they have been remapped to.

See Tableau Services Manager Ports.

Local firewall configuration

If you are running a firewall on the computer where you will be installing Tableau Server, then you will need to open the following default ports for Tableau Server traffic. All port numbers, except 443 can be changed.
<table>
<thead>
<tr>
<th>Port</th>
<th>TCP/UDP</th>
<th>Used by ...</th>
<th>TYPE OF INSTALLATION</th>
</tr>
</thead>
<tbody>
<tr>
<td>80</td>
<td>TCP</td>
<td>Gateway</td>
<td>X</td>
</tr>
<tr>
<td>443</td>
<td>TCP</td>
<td>SSL. When Tableau Server is configured for SSL, the application server redirects requests to this port. Do not change this port.</td>
<td>X</td>
</tr>
<tr>
<td>8850</td>
<td>TCP</td>
<td>Tableau Services Manager.</td>
<td>X</td>
</tr>
<tr>
<td>8060</td>
<td>TCP</td>
<td>PostgreSQL database.</td>
<td>X</td>
</tr>
<tr>
<td>8061</td>
<td>TCP</td>
<td>PostgreSQL backup verification port</td>
<td>X</td>
</tr>
<tr>
<td>8000-9000</td>
<td>TCP</td>
<td>Range of ports reserved by default for dynamic mapping of Tableau processes</td>
<td>X</td>
</tr>
<tr>
<td>27000-27009</td>
<td>TCP</td>
<td>Range of ports used by Tableau Server for License service. This range must be open on the node running the License service and accessible from other nodes. By default, the initial node runs the License service.</td>
<td>X</td>
</tr>
</tbody>
</table>

See Tableau Services Manager Ports and Configure Local Firewall.

**System user and groups**

Tableau Server on Linux uses one unprivileged user, and two groups for proper operation. Tableau will create the default account and groups during setup. Alternatively, you can specify existing accounts. See System user and groups.

**Sudo and root access**
All installation tasks and administrative tasks for Tableau Server must be run as root. Often this is accomplished using the sudo command, but running the commands directly as the root user is also possible.

To install Tableau Server with the root account, you must specify a user account during installation. The account will be used for managing TSM. Specify the account by running the initialize-tsm script with the \texttt{-a} option. See Help Output for initialize-tsm Script.

**Account password**

The user account that you use to install and administer Tableau Server must be able to authenticate with a password. That is, the user must not use another means of authenticating (for example public key authentication).

If the account you are using to install and initialize Tableau Server does not have a password, you can set one using the \texttt{passwd} command:

\begin{verbatim}
sudo passwd $USER
\end{verbatim}

**Port access requirements**

If you want to install Tableau Server remotely, for example by means of SSH, ensure that the following ports are open:

- \texttt{8850}. The port used for the Tableau Services Manager (TSM) web interface. You can use this interface to configure Tableau Server.
- \texttt{80}. The port used for the main Tableau Server web interface.

The Tableau Server installer attempts to open these ports during the installation process, but it can only open these ports for the \texttt{firewalld} firewall. If your computer runs another firewall, you must open the ports before you install.

**Virtual Container environments**
Installation of Tableau Server in virtual container environments such as Docker have not been tested and are not supported. Tableau Server on Linux will not function as expected if installed in these environments.

Package requirements

**Systemd**

Tableau Server requires systemd to manage services. This package is installed by default on CentOS 7 and Ubuntu 16. If you decide to test Tableau Server on a modified version of these distributions, you can run the following command to confirm that systemd is installed:

```
whereis systemd
```

If systemd is installed, the installation location is displayed. For example, you might see the following output:

```
systemd: /usr/lib/systemd /etc/systemd /usr/share/systemd /usr/share/man/man1/systemd.1.gz
```

If you have systemd installed but the Tableau installer is failing requirements checks for systemd, it's likely that systemd is not running. To verify that systemd is running, run the following command:

```
ls /run/systemd
```

The output will be a list of files and directories. If systemd is running, the output will include system. If system is not in the output, then systemd is not running.

*Continue to the next step: Install and Configure Tableau Server.*
Minimum Hardware Requirements and Recommendations for Tableau Server

The following minimum hardware requirements and recommendations apply to all computers running Tableau Server, including physical hardware and virtual machines (VMs):

- **Minimum requirements for testing and prototyping** are the minimum hardware your computer must have in order to install Tableau Server. If your computer does not meet these requirements, the Setup program will not install Tableau Server. These requirements are appropriate for testing and prototyping, but are not appropriate for most production environments.

- **Minimum recommendations for production** go beyond minimum requirements, and represent the minimum hardware configuration you should use for a production installation of Tableau Server. If your computer meets the minimum requirements but does not meet these recommendations, the Setup program will warn you but you can continue the installation.

In addition, Tableau Server should not be installed on a physical computer or on a VM instance that is also running resource-intensive applications such as databases or application servers.

**Note:** If you install Tableau Server on a computer that meets the minimum requirements but does not have at least 8 cores and 16 GB of system memory, the default number of all processes installed is reduced to one of each process by design. For more information about processes, see Server Process Limits.

Minimum Hardware Requirements for Testing and Prototyping

The computer on which you are installing or upgrading Tableau Server must meet the minimum hardware requirements. If the Setup program determines that your computer does not meet the following requirements, you will not be able to install Tableau Server.

These minimum requirements are appropriate for a computer that you use for prototyping and testing of Tableau Server. They apply to single-node installations and to each computer in a distributed installation.

<table>
<thead>
<tr>
<th>Processor CPU</th>
<th>RAM Free</th>
</tr>
</thead>
<tbody>
<tr>
<td>64-bit</td>
<td>16 GB</td>
</tr>
</tbody>
</table>

**Minimum Hardware Requirements**

**Note:** These minimum requirements are not recommended for use in production environments. For production minimum recommendations, see Minimum Hardware Recommendations.

For the requirements:

- Free disk space is calculated after the Tableau Server Setup program is unzipped. The Setup program uses about 1 GB of space. You may need to allocate additional disk space depending on various factors like whether you will be using extracts. For more information, see Disk Space Requirements.

The core Tableau Server bits must be installed in a directory with at least 15 GB of free disk space. If you attempt to install Tableau Server on a computer that does not have enough space, the Tableau Server package will install, but you will be unable to continue with setup. By default the install location is the `/opt` directory. You can change the installation path for Tableau Server on RHEL distros. For more information, see Option to install to non-default location on RHEL-distros.
If you plan to make heavy use of extracts then you may need to allocate additional disk space. You can specify a different directory for data (extract) storage during installation. See Install and Initialize TSM.

- Core count is based on "physical" cores. Physical cores can represent actual server hardware or cores on a virtual machine (VM). Hyper-threading is ignored for the purposes of counting cores.

**Minimum Hardware Recommendations**

For production use, the computer on which you install or upgrade Tableau Server should meet or exceed the minimum hardware recommendations. These recommendations are general. Actual system needs for Tableau Server installations can vary based on many factors, including number of users and the number and size of extracts. If the Setup program determines that your computer does not meet the following recommendations, you will get a warning, but you can continue with the setup process. The minimum recommendations listed here are intended as general guidance. However the recommendations for your environment may vary. For more information, see the Hardware recommendations section of the Recommended Baseline Configurations topic.

<table>
<thead>
<tr>
<th>Install Type</th>
<th>Processor</th>
<th>CPU</th>
<th>RAM</th>
<th>Free Disk Space</th>
</tr>
</thead>
<tbody>
<tr>
<td>Single node</td>
<td>64-bit</td>
<td>8-core, 2.0 GHz</td>
<td>32 GB</td>
<td>50 GB</td>
</tr>
<tr>
<td></td>
<td></td>
<td>or higher</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

If you are adding Tableau Prep Conductor to your Tableau Server installation, we recommend you add a second node and dedicate this to running Tableau Server Prep Conductor. This node should have a minimum or 4 cores (8 vCPUs), and 16 GB of RAM.

**Multi-node and enterprise deployments** Contact Tableau for technical guidance.

Nodes must meet or exceed the minimum hardware recommendations, except:

- Nodes running backgrounder, where 4 cores may be acceptable.
For the recommendations:

- Free disk space is calculated after the Tableau Server Setup program is unzipped. The Setup program uses about 1 GB of space. You may need to allocate additional disk space depending on various factors like whether you will be using extracts. For more information, see Disk Space Requirements.

The core Tableau Server bits must be installed in a directory with at least 15 GB of free disk space. If you attempt to install Tableau Server on a computer that does not have enough space, the Tableau Server package will install, but you will be unable to continue with setup. By default the install location is the /opt directory. You can change the installation path for Tableau Server on RHEL distros. For more information, see Option to install to non-default location on RHEL-distros.

If you plan to make heavy use of extracts then you may need to allocate additional disk space. You can specify a different directory for data (extract) storage during installation. See Install and Initialize TSM.

- Core count is based on "physical" cores. Physical cores can represent actual server hardware or cores on a virtual machine (VM). Hyper-threading is ignored for the purposes of counting cores.

- RAM shown is the minimum recommended for a single-node installation. Your installation may function better with more RAM, depending on activity, number of users, and background jobs, for example. For hardware specifications Tableau uses internally for testing scalability, see Hardware Recommendations.

For hardware recommendations for Tableau Server in the cloud, see the following:
Install and Initialize TSM

This topic guides you through the process of installing Tableau Server and initializing Tableau Services Manager (TSM).

**Important:** Do not install a beta version of Tableau Server in your production environment. You should also never restore a production Tableau Server installation using a backup of a beta version.

Prerequisites

Before proceeding, review the topic, Before you install....

Review optional initialization parameters

Before you install and initialize TSM, it is critical that you review the parameters that you may optionally set as part of the initialization operation. You can only run `initialize-tsm` once, so be sure to run it with all of the options that your organization needs. Some options, such as non-default system user and group can only be configured as part of the initialization operation. Other configurations, such as forward proxy and dynamic port settings can be manually set after you run initialization, but doing so is a much more labor-intensive process.

For a complete list of optional parameters, see Help Output for initialize-tsm Script.

Some common scenarios where optional initialization parameters are used:
Configuring Tableau Server to work with a forward proxy server. See Optional: common initialize-tsm parameters, later in this topic, to configure Tableau Server during installation. You can also configure Tableau Server after you install, see Configuring Tableau Server on Linux to work with a forward proxy.

Specifying dynamic port mapping. By default, most ports needed by Tableau Server are assigned (mapped) dynamically from a predefined range of ports. The port assignments are made for each service or process during installation. If you want to modify port mapping, we recommend configuring this during installation, see Controlling port remapping with initialize-tsm.

Specifying non-default system user or group. This configuration change can only be performed during initialization. See System user and groups.

Specifying a non-default data directory. This configuration change can only be performed during initialization. See Data directory.

Install Tableau Server

Install Tableau Server with your distribution’s package manager, then run a script to initialize TSM. The script is included with the installed package.

**Important:** The hostname of the server must not change after you start TSM. For example, this might happen if you use the cloud-init package to initialize a virtual machine, and you install Tableau Server on that virtual machine.

Install the Tableau Server package

By default, Tableau Server is installed in the /opt directory. On RHEL-like distributions you can specify a different install location.

1. Log on as a user with sudo access to the computer where you want to install Tableau Server.

2. Download the .rpm or .deb installer package from the Tableau Server Downloads and Release Notes page.

3. Navigate to the directory where you copied the .rpm or .deb package.
4. Use the package manager to install the Tableau Server package.

- On RHEL-like distributions, including CentOS, you have the option to install Tableau to a non-default location.

  - **Default location**—To install to the default location (/opt/tableau/tableau_server), run the following commands, where `<version>` is formatted as major-minor-maintenance (ex: 2019-2-5):

    ```
    sudo yum update
    sudo yum install tableau-server-<version>.x86_64.rpm
    ```

  - **Non-default location**—To install to a non-default location, you must use `rpm -i`. You will also need to install all dependent packages. See the note below.

    Run the following command:

    ```
    sudo rpm -i --prefix /preferred/install/path tableau-server.rpm
    ```

**Note:** When you use `yum` to install Tableau Server, all dependent packages are automatically downloaded and installed. This is the preferred method for installing Tableau. If you want to install to a non-default location, or your organization does not allow you to use `yum` and you must install using `rpm -i`, you must also install all dependent packages separately. For information about installing dependent packages, see Installing Tableau Server on an Air-Gapped Computer Running Linux.

- On Ubuntu and Debian, run the following commands, where `<version>` is formatted as major-minor-maintenance (ex: 2019-2-5):

```
sudo apt-get update
sudo apt-get upgrade
sudo apt-get -y install gdebi-core
sudo gdebi -n tableau-server-<version>_amd64.deb

Initialize TSM

You can specify a separate location for data (extracts and extract metadata) storage, whether or not you install Tableau to the default location.

1. Navigate to the scripts directory:
   
   cd /opt/tableau/tableau_server/packages/scripts.<version_code>/

2. Run the following script to start TSM:
   
   sudo ./initialize-tsm --accepteula --<optional_parameters>

   The only required parameter for the initialize-tsm script is --accepteula. You must include this parameter to accept the Tableau Server End User License Agreement (EULA). The EULA is available in the following location:

   /opt/tableau/tableau_server/packages/docs.<version_code>/Commercial_EULA.txt

   However, review the following common parameters before running the script.

Optional: common initialize-tsm parameters

There are a number of parameters (also referred to as, flags) that you can set when you run initialize-tsm. Common flags are listed below. For a complete list, run the command, sudo ./initialize-tsm -h, or see Help Output for initialize-tsm Script.
The -a flag to specify a user to be added to the tsmadmin and tableau groups instead of the user running this script. If you are installing with the root account, you must specify the -a flag.

The --unprivileged-user flag to set a different service account. By default a new user called tableau will be created. This account is an unprivileged service account under which most Tableau process run. We recommend creating a different user only in the case where a tableau user account already exists on the computer.

The -d flag to specify a non-default location of the "data directory" where Tableau Server stores extracts, information about extracts, and more.

By default, Tableau Server uses the following location for the directory:
/var/opt/tableau/tableau_server

When you set this flag, the initialize-tsm script will create and apply permissions to the directory that you specify. There are important restrictions that apply to changing the default directory path. See Data directory.

The --debug flag for troubleshooting. This option displays each command in the script as it is run and can make it easier to troubleshoot issues. Use of this option results in extensive output to the screen.

We recommend configuring Tableau Server for a forward proxy solution during installation.

To do so, include the --http_proxy and/or --https_proxy flags to specify the forward proxy server. Specify the URL with the port, for example:

--http_proxy=http://proxy.example.lan:80/
--https_proxy=http://1.2.3.4:443/
Take care to use http when you specify the URL for the https_proxy environmental variable. Do not specify the https protocol for the value of the https_proxy environmental variable.

To configure Tableau Server to bypass the forward proxy, include the --no_proxy flag. You should also add exceptions to this proxy configuration to guarantee that all communications within a local Tableau Server cluster (if you have one now or will have one later) do not route to the proxy server. For example:

```
--no_proxy=
y=localhost,127.0.0.1,localaddress,.localdomain.com.
```

- If you want to manually manage port assignment for TSM and Tableau Server processes, you may need to use one or more port-related switches with initialize-tsm. For more information, see Controlling port remapping with initialize-tsm.

3. Log off and log on again to the terminal before you configure Tableau Server.

When you log on again, you create a new session in which group membership changes have taken effect. The new session also has access to the environment variables added by the initialize-tsm script.

Alternatively, you can run the following command to update your path for the current session (but not to update your group membership):

```
source /etc/profile.d/tableau_server.sh
```

Next step

- Activate and Register Tableau Server

**Activate and Register Tableau Server**

This topic describes how to activate and register Tableau Server.
Tableau Server on Linux Administrator Guide

Tableau Server requires at least one product key that both activates the server and specifies the number of license levels you can assign to users. You can access your product keys from the Customer Portal.

If you need to activate Tableau Server on a computer that is offline, see Activate Tableau Server Offline - Adding a License. If you need to activate additional product keys to add capacity to an existing Tableau Server installation, see Add Capacity to Tableau Server.

Prerequisite

Before proceeding with the procedures in this topic, complete the following prerequisites as outlined in Install and Configure Tableau Server:

- Install and Initialize TSM
- TSM uses port 8850. If you are running a local firewall, open port 8850. See Configure Local Firewall.

Use the TSM web interface

1. Sign in to Tableau Services Manager Web UI.

   The account you use must have administrative privileges on the computer where TSM is installed.

2. On the Activate page, Enter or paste your product key and click Activate License.
3. On the Register page, enter your information into the fields and click **Register**.
Use the TSM CLI

Activate Tableau Server

To activate Tableau Server for production use, you must have a product key.

To activate a product key, run the following command:

tsm licenses activate -k <product key>

To activate a two-week trial, run the following command:

tsm licenses activate -t

If you are unable to activate Tableau, for example, if you get an error like this:

License Server not available

see Activate Tableau Server Offline - Adding a License.

What if I get an access denied error when I attempt to run TSM commands?

The account that you use to configure the rest of the installation must be a member of the tsmadmin group that was created during initialization. To view the user accounts in the tsmadmin group, run the following command:

grep tsmadmin /etc/group

If the user account is not in the group, run the following command to add the user to the tsmadmin group:

sudo usermod -G tsmadmin -a <username>

Register Tableau Server

After activation, you need to register Tableau Server. To do this, create a registration file and then pass it as an option with the tsm register command.
1. Generate a template that you can edit by running the following command:

```
  tsm register --template > /path/to/<registration_file>.json
```

2. Edit the template file to create your completed registration file.

Here is an example of a registration file in the required format:

```
{
  "zip" : "97403",
  "country" : "USA",
  "city" : "Springfield",
  "last_name" : "Simpson",
  "industry" : "Energy",
  "eula" : "yes",
  "title" : "Safety Inspection Engineer",
  "phone" : "5558675309",
  "company" : "Example",
  "state" : "OR",
  "department" : "Engineering",
  "first_name" : "Homer",
  "email" : "homer@example.com"
}
```

3. After saving changes to the file, pass it with the --file option to register Tableau Server:

```
  tsm register --file /path/to/<registration_file>.json
```

For example:

```
  tsm register --file /usr/share/tableau-reg-file.json
```

If you have a product key for Add-on license, you must activate that product key to be able to use the features included in each Add-on.
Next step

- Configure Initial Node Settings

Offline Activation When Installing Tableau Server

When you install Tableau Server, you have to activate at least one product key. Doing this activates the server, and specifies the number of license levels you can assign to users. There are also times you may need to activate licenses after Tableau Server is installed, for example, if you add capacity to your server, or get a new product key. If you don't have your product key, you can get it from the Tableau Customer Account Center.

In most cases, you can activate your key directly from Tableau, either during installation, or later, using the Tableau Services Manager (TSM) Licenses page, but there are some circumstances that don't allow you to do this. If your computer is not connected to the internet for example, or has a firewall that restricts access outside your intranet. In these cases you need to do an offline activation.

- Offline activation when adding a license—To complete an offline activation when you are adding a license to Tableau Server, see Activate Tableau Server Offline - Adding a License.
- Offline activation of licenses during install—To complete an offline activation when you are installing Tableau Server, follow the steps below.

Offline activation overview and prerequisites

Offline activation of Tableau Server involves several steps. If you have never activated a product key before on your computer, you need to run through the steps twice, once to initialize licensing on the computer, and a second time to activate the key. If you have had a license on the computer in the past, you run through the steps only once.
Offline activation steps:

1. Generate an offline activation request file (`offline.tlg`).
2. Upload the offline activation request to Tableau and download the resulting activation file (`activation.tlf`).
3. Upload the activation file to Tableau Server.

If this is the first time a Tableau product key has been activated on the computer, you repeat these three steps a second time.

Use the TSM web interface

If you are unable to do an activation of your product key from within the TSM Setup UI, you can do an offline activation. This might be necessary if you are installing on a machine that is isolated without internet access. If you attempt to activate your product key and see a dialog that says online activation is unavailable, you can activate the key offline. Click **Activate License Offline**.
Instructions for a five-step process display.

1. Step 1: Create Offline File

Create an offline file you will upload to Tableau. If your license key is not pre-filled in the form, enter your key and click Create Offline File to generate an offline file (offline.tlq) on the local computer.
Leave the Tableau Server install program open and copy the offline.tlq file to a computer with internet access. You need to upload this file to Tableau to generate an activation file.

2. Step 2: Upload and Submit Offline File

On the computer where you copied the offline.tlq file, open a browser and go to http://www.tableau.com/support/activation to open the Tableau Support Activation page. You will upload and submit the offline.tlq file to Tableau. This automatically generates an activation file (activation.tlf) that you can download and copy back to the Tableau Server computer.

   a. On the Offline Activation page, click **Choose File** to select the offline.tlq file.
   b. Click **Upload Activation File** to submit the file to Tableau.
   c. Click the **here** link to download the activation.tlf file to your computer.

**Offline Activation**

The activation was successful. Please click **here** to download your activation file.

For help creating the offline activation file, see [Activate Tableau Desktop Offline](https://www.tableau.com/support/activate) or [Activate Tableau Server Offline](https://www.tableau.com/support/activate) (Linux)

d. Copy the downloaded activation.tlf file to the computer where Tableau Server is installed.

3. Step 3: Upload Activation File

On the Tableau Server computer, click **Upload Activation File** to upload the file to Tableau Server.
4. **Step 4: Create, Upload and Submit a Second Offline File**

   Do the first two steps again, to create a new `offline.tlq` file, submit it to Tableau, and download another `activation.tlf` file.

5. **Step 5: Upload Second Activation File**

   On the Tableau Server computer, upload the second `activation.tlf` file. When you do this successfully, the **Activate License** option is enabled.

6. Click **Activate License** to complete the offline activation and continue with the installation process.

**Use the TSM CLI**

Log in to Tableau Services Manager

Before you can proceed you must log in to Tableau Services Manager (TSM).

To log in to TSM, run the following command:
tsm login -u <username>

What if I can't log in?

The account that you use to configure the rest of the installation must be a member of the tsmadmin group that was created during initialization. To view the user accounts in the tsmadmin group, run the following command:

grep tsmadmin /etc/group

If the user account is not in the group, run the following command to add the user to the tsmadmin group:

sudo usermod -G tsmadmin -a <username>

After you have added the user to the tsmadmin group, run the tsm login command.

Step 1 - Generate an offline activation request file

1. On the initial node, open a terminal session.

2. Type this command to get your offline activation file:

   tsm licenses get-offline-activation-file -k <product-key> -o <target-directory>

   You can get your product key from the Tableau Customer Portal. The target directory must already exist.

3. Copy the offline activation file (offline.tlq) from the target directory to a computer that has internet access.

Step 2 - Upload the offline activation request to Tableau

1. On the computer that has internet access, go to the Tableau Product Activations page.

2. Complete the instructions to upload your offline.tlq file.
This creates an activation file, activation.tlf.

3. Download the resulting activation file from Tableau.

Step 3 - Initialize or activate your license

1. Copy the activation file (activation.tlf) to a location accessible from your Tableau Server computer.

2. Run the following command:

   tsm licenses activate -f <path-and-activation-file>

3. Restart the server for licensing changes to take effect:

   tsm restart

First time license activation

If this is the first Tableau license activated on the computer, you will see this message:

Your license has been initialized. To complete the activation, we need one more exchange. Generate and send to Tableau a second activation request file.

Repeat steps 1 through 3 above to activate your license. You need to generate a second offline.tlq file, upload it to Tableau, and download the resulting activation.tlf file, then use that file to activate your license.

If you are not sure whether you need to repeat the steps, you can run this command:

   tsm licenses list

If you see a message like this one you need to repeat the steps above:

   No licenses are currently activated.

Subsequent license activation

If this is not the first Tableau license activated on the computer, or if you have completed the steps above two times, you should see a success message:
Activation successful.

Tableau Server is activated. If you need additional assistance, contact Tableau Technical Support.

Deactivate Tableau Server Offline

If Tableau Server does not have Internet access, you can use the instructions in this topic to deactivate Tableau Server. You will need to use a second computer that is able to access the Internet to complete this activation process.

**Note:** These instructions require you to save a file that you will use for deactivation. You must complete these steps using a browser other than Microsoft Edge, such as Google Chrome. If you use Edge, the product key return file used for deactivation is not created.

1. On the Tableau Server initial node, create a directory to store the offline deactivation file that is created in the next step.

2. Log in to TSM using a Tableau Administrator account, and then run the following command:

   ```
   tsm licenses get-offline-deactivation-file -k <productkey> -o <deactivation-file-directory>
   ```

3. Move the `deactivate.tlq` file from the deactivation file directory that you specified in the previous command to a trusted computer that has Internet access.

4. On the trusted computer that has Internet access, open a web browser and visit the Tableau Product Activations page. Follow the instructions on that page to submit your `deactivate.tlq` file.

5. When prompted, save the product key return file (`return.tlr`) from the Product
6. Move the product key return file \texttt{(return.tlr)} from the trusted computer that has Internet access to the Tableau Server initial node that runs the Licensing Server service.

7. Log in to TSM using a Tableau Administrator account, and then run the following command:

\begin{verbatim}
 tsm licenses deactivate -f <path-to-license-key-return-file>
\end{verbatim}

Configure Initial Node Settings

This topic describes how to configure essential server settings as part of the initial Tableau Server installation process.

Prerequisite

Before proceeding with the procedures in this topic, complete the following prerequisites as outlined in Install and Configure Tableau Server:

- Install and Initialize TSM
- Activate and Register Tableau Server

You may also need to configure your local firewall for Tableau Server traffic. See Configure Local Firewall.

Use the TSM web interface

After you have activated and registered Tableau Server, the installation program will display the Setup page.
**Note:** If you need to configure Tableau Server to connect to an LDAP directory that is not Active Directory, then you must use the TSM CLI.

<table>
<thead>
<tr>
<th>Identity Store</th>
</tr>
</thead>
<tbody>
<tr>
<td>You cannot change the identity store after initializing.</td>
</tr>
<tr>
<td>- Local</td>
</tr>
<tr>
<td>- Active Directory</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Gateway Port</th>
</tr>
</thead>
<tbody>
<tr>
<td>Port Number: 80 (Default)</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Product Usage Data</th>
</tr>
</thead>
<tbody>
<tr>
<td>- Disable sending usage data to Tableau</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Include samples</th>
</tr>
</thead>
<tbody>
<tr>
<td>- Include sample workbooks</td>
</tr>
</tbody>
</table>

[Initialize]
Identity store settings

You must configure the identity store settings for the Tableau Server computer. The identity store manages Tableau Server accounts. You can configure the identity store to synchronize with an external directory (for example, OpenLDAP or Active Directory) or you can configure the identity store to manage and store accounts on Tableau Server. If you will be using a single sign on solution (OpenID, SAML, Kerberos, etc) then review the following topics before configuring the identity store:

- Identity Store
- Authentication

**Important:** After you have configured and applied settings for the identity store, it cannot be changed.

If you select **Active Directory**, Tableau Server will populate the **Domain** and **NetBIOS** fields from the computer on which you are running Setup. In some cases, Setup may not display these attributes. For more information about how Tableau Server connects and communicates with Active Directory, see User Management in Active Directory Deployments.

Tableau Server requires read access to Active Directory. You can use simple bind or GSSAPI bind to authenticate Tableau Server with Active Directory.

**LDAP simple bind**
If you are using simple bind to authenticate with Active Directory, enter a domain account and password. We recommend using LDAPS to connect. See LDAP over SSL.

**LDAP GSSAPI bind**
Identity Store

You cannot change the identity store after initializing.

- Local
- Active Directory

Domain: example.lan
 NetBIOS (Nickname): example

Hostname (Optional)  Port (Optional)

Tableau Server requires read access to Active Directory. Specify how Tableau Server will authenticate with Active Directory.

- LDAP simple bind
- LDAP GSSAPI bind

Specify a user principal name (UPN) and upload the Kerberos configuration file Tableau Server will use to authenticate to the Identity Store.

UPN
tabserv@example.lan

Configuration file

Specify and configure the method Tableau Server will use to authenticate to Active Directory.

- Keytab file
- Username and password

Keytab file

Username  Password
We recommend binding to LDAP directory with GSSAPI. To bind with GSSAPI you will need a keytab file specifically for the Tableau Server service. See Understanding Keytab Requirements.

Gateway port

The default port for web access to Tableau Server (via HTTP) is port 80. If the installation program determines that port 80 is in use when you first install Tableau Server, an alternate port (for example 8000) is used and shown in the Port number box.

You may need to change the port for other networking needs, for example, if you have a hardware firewall or proxy in front of the Tableau Server host, this might make running a back-end system on port 80 undesirable.

Product usage data

By default, Tableau Server shares usage data with Tableau that helps us better understand how you use our products, improve your overall experience, and build highly intelligent features that make Tableau even more powerful.

Clear this option if you do not want usage data sent to Tableau.
You can also change this setting after installation, on the TSM Maintenance tab in the TSM Web UI, or using the TSM CLI. For more information, see Server Usage Data.

Sample workbook installation

By default, Tableau Server will install sample workbooks in the Default site when you initialize the server.

Alternatively, you can publish samples after installation by using the publishsamples tabcmd command.

Initializing install

After you have configured the options on this page, click **Initialize**.

The initialization process can take a while. When initialization is complete the following page is displayed:
Use the TSM CLI

First, configure identity store, gateway settings, and sample workbook installation. Then apply the changes, optionally verify your LDAP connection, and then initialize Tableau Server.

Configure identity store settings

You must configure the identity store settings for the Tableau Server computer.

**Important:** After you have configured and applied settings for the identity store, it cannot be changed.

Use the json template in identityStore Entity to create a json file. After you have filled in the options with the appropriate values, you can then pass the json file with this command:

```
tsm settings import -f path-to-file.json.
```
Configure gateway settings (optional)

Depending on your network requirements, you may need to configure the gateway settings for the Tableau Server computer. For example, if you are enabling SSL or configuring access to Tableau Server with a reverse proxy, you may need to configure gateway settings. See gatewaySettings Entity for more information.

Use the json template in gatewaySettings Entity to create a json file. After you have filled in the options with the appropriate values, you can then pass the json file with this command:

`tsm settings import -f path-to-json-file.json`

Configure product usage data (optional)

By default, Tableau Server shares usage data with Tableau to help us better understand how you use our products. This allows us to improve your overall experience and build highly intelligent features that make Tableau even more powerful.

Tableau collects only behavioral and usage data, never any of your confidential database values, and your usage data will never be shared or sold; its sole purpose is to improve your Tableau experience.

If you do not want to share product usage data, use the json template in shareProductUsageDataSettings Entity to create a json file, and specify a value of false. Then pass the json file with this command:

`tsm settings import -f path-to-json-file.json`

You can also change this setting after installation, on the TSM Maintenance tab or using the TSM CLI. For more information, see Server Usage Data.

Configure sample workbook installation (optional)

By default, Tableau Server will install sample workbooks in the Default site when you initialize the server.
Tableau Server on Linux Administrator Guide

If you do not want to install sample workbooks during installation, run the following command:

tsm configuration set \-k install.component.samples \-v false

You can publish samples after installation by using the publishsamples tabcmd command.

Apply pending configuration changes

Now that you've created and set initial configuration, you must apply them. When you apply configuration changes, tsm will verify the settings you've set before committing them.

To apply configuration changes to Tableau Server, run this command:

tsm pending-changes apply

The `pending-changes apply` command displays a prompt to let you know this will restart Tableau Server if the server is running. The prompt displays even if the server is stopped, but in that case there is no restart. You can suppress the prompt using the `--ignore-prompt` option, but this does not change the restart behavior. For more information, see tsm pending-changes apply.

Once this command has completed, TSM processes are running, and Tableau Server is configured but is not running.

Verify LDAP configuration (Optional)

If your identity store uses Active Directory or OpenLDAP, then we recommend verifying LDAP connectivity before proceeding.

To do so, run the following commands before you initialize the server:

tsm user-identity-store verify-user-mappings \-v <user name>

tsm user-identity-store verify-group-mappings \-v <group name>

User and group names must be valid names that exist in the LDAP server that you are connecting to. If your LDAP connection is set up correctly, then the user or group attributes will
be returned to the shell. If your connection is not set up correctly, then an error will be returned.

Initialize and start Tableau Server

- To initialize and start Tableau Server, use the `--start-server` option:

  ```bash
  tsm initialize --start-server --request-timeout 1800
  ```

  This saves time by starting the server running after initialization.

- If you are going to reconfigure Tableau Server after initialization, leave the `--start-server` option off:

  ```bash
  tsm initialize --request-timeout 1800
  ```

  This stops the server after initialization.

Start Tableau Server. If you did not use the `--start-server` option during initialization and are finished configuring Tableau Server, use this command to start the server:

```bash
tsm start --request-timeout 900
```

**Note:** If you experience timeouts when installing or configuring Tableau Server, you may need to specify a longer timeout. For more information, see Install fails due to timeouts.

Next Step

After initialization is complete, create the Tableau Server administrator user account. See Add an Administrator Account.
Configuration File Example

This article provides an example of a complete JSON configuration file, with gatewaySettings and identityStore entities specified. In addition, a configuration key sets the gateway timeout to 900 seconds.

Your configuration file will look different depending on the options you need to set.

You might set multiple .json configuration files during installation. To set the values for each file in Tableau Server, you run the following command, once for each configuration file:

```bash
tsm settings import -f path-to-file.json
```

After you set the configuration files, run tsm pending-changes apply to apply the changes from all of the .json files you've set.

```json
{
  "configEntities": {
    "gatewaySettings": {
      "_type": "gatewaySettingsType",
      "port": 80,
      "firewallOpeningEnabled": true,
      "sslRedirectEnabled": true,
      "publicHost": "localhost",
      "publicPort": 80
    },
    "identityStore": {
      "_type": "identityStoreType",
      "type": "local",
      "domain": "example.lan",
      "nickname": "EXAMPLE"
    }
  },
  "configKeys": {
    "gateway.timeout": "900"
  }
}
```
Entities vs keys

As shown in the example above, there are two classes of configuration parameters: configEntities and configKeys.

**configEntities**

Certain types of configuration are done through entity sets that map to specific scenarios, such as the identity store and gateway configurations. When you pass a set of configEntities with the `tsm settings import -f path-to-file.json` command, TSM validates the configuration. If values passed are invalid, TSM will provide an error. This enables you to make changes during the configuration process, rather than experience a configuration failure at initialization or run time.

Entities can be set only by including a configEntities block in a .json file.

**Important:** All files that are referenced in configEntities must be located on the local computer. Do not specify UNC paths.

**configKeys**

Entities cover only a small portion of the configuration values that can be set. Hundreds of keys correspond to parameters stored in .yaml files. Tableau Server uses these parameters to store all of the configuration information for all services.

You can set individual keys with the `tsm configuration` command. But during deployment, setting them along with other configuration scenarios in JSON files, as shown above, is more convenient.

Unlike configEntities, configKeys are not validated.
Server Usage Data

The Tableau Server administrator can control whether or not usage data from Tableau Server is sent to Tableau. By default this option is enabled, and can be disabled at initial install, or after installing Tableau Server, using the TSM Web UI or command line.

Usage data from Tableau helps us better understand how you use our products. This gives us information to improve your overall experience and build highly intelligent features that make Tableau even more powerful.

Tableau collects only behavioral and usage data. A sample of collected data may include how long users engage with certain features, the paths they take in the product, where they click, and what questions they ask in Ask Data. Your usage data will be handled according to the Tableau Privacy Policy.

Disable Usage Data Sharing

You can disable the sharing of usage data when you install Tableau Server, or at any time after installation.

Disabling the sharing of usage data at install

To disable the sharing of usage data with Tableau when you are installing Tableau Server, clear the option during the initial configuration of server. For details, see Product usage data.

Disabling the sharing of usage data after install

Use the TSM web interface

1. Open TSM in a browser:

   https://<tsm-computer-name>:8850. For more information, see Sign in to Tableau Services Manager Web UI.
2. Click the **Maintenance** tab.

3. Under Other Maintenance Tasks, in Server Usage Data, clear **Send usage data to improve Tableau features**:

   ![Product Usage Data](image)

   **Help us improve your Tableau experience by sharing how you use the product. Tableau collects information about your feature usage. All usage data will be handled according to our Privacy Policy - [http://www.tableau.com/privacy](http://www.tableau.com/privacy)**

   ![Learn more](image)

   - Disable sending usage data to Tableau

4. When you are finished, click **Pending Changes**, and then click **Apply Changes and Restart**.

Use the TSM CLI

If you do not want to share product usage data, disable the option using this tsm configuration command:

```bash
$ tsm configuration set -k shareproductusagedata.enabled -v false
$ tsm pending-changes apply
```

The `pending-changes apply` command displays a prompt to let you know this will restart Tableau Server if the server is running. The prompt displays even if the server is stopped, but in that case there is no restart. You can suppress the prompt using the `--ignore-prompt` option, but this does not change the restart behavior. For more information, see `tsm pending-changes apply`. 
Add an Administrator Account

The final step in activating Tableau Server is to add the initial administrator account. The administrator will have all access to the server including the ability to manage users, groups, and projects.

The server must be running when you create the initial admin user.

If you have configured the Tableau Server identity store to use LDAP or Active Directory, then the initial administrative user that you specify must be an account in the directory. The initial administrative user is generally distinct from the user account on the Tableau Server computer that you use to run tsm.

However, these accounts can be the same if you have configured the Tableau Server identity store to use LDAP or Active Directory and the initial admin user is a member of the tsmadmin group on the Tableau Server computer.

Prerequisites

Before proceeding with the procedures in this topic, complete the following prerequisites as outlined in Install and Configure Tableau Server:

- Install and Initialize TSM
- Activate and Register Tableau Server
- Configure Initial Node Settings

Use web UI

After Tableau Server is finished initializing, the installation program will display a page to create the Tableau Server administrator.

- If you configured a local identity store during setup, then specify a name and password that you want to use.
If you configured a LDAP or Active Directory identity store during setup, then you must specify a user account that is a member of the directory.

If you are installing remotely, then you must sign in to TSM on the physical computer where Tableau Server is installing, or you can access the computer remotely and run the tabcmd initialuser command from a shell.

**Use tabcmd CLI**

You must create the initial administrative account for Tableau Server.

- If you configured a local identity store during setup, then specify a name and password that you want to use.

- If you configured a LDAP or Active Directory identity store during setup, then you must specify a user account that is a member of the directory.

To create the initial user, run the following `tabcmd` command:

```
tabcmd initialuser --server http://localhost --username '<new-admin-username>'
```

For example:

```
tabcmd initialuser --server http://localhost --username 'tableau-admin'
```

If you are running the HTTP protocol on a port other than 80, specify the port after the host name, for example: `--server http://localhost:8080`.

After you run the command, the shell will prompt for an administrative password.

**Next steps**

After you have created the Tableau Server administrator account, continue your deployment by working through the configuration topics at Post Installation Tasks.
Important: You must install the PostgreSQL driver if you want to use the built-in administrative views. You can find driver links and installation instructions for all the supported connectors on the Driver Download page.

Validate Installation

To validate that Tableau Server is installed and running properly and to review the built-in administrative views, you must install the PostgreSQL driver.

Prerequisites

Before proceeding with the procedure in this topic, complete the following prerequisites as outlined in Install and Configure Tableau Server:

- Install and Initialize TSM
- Activate and Register Tableau Server
- Configure Initial Node Settings
- Add an Administrator Account

Install PostgreSQL driver and validate installation

To validate installation:

1. Download PostgreSQL drivers from the Driver Download page.

2. Install the drivers on your Tableau Server computer:

   - On CentOS and RHEL, download the .rpm file and then run the following command:

     ```
     sudo yum install <driver-installation-package>.rpm
     ```

     For example:
sudo yum install tableau-postgresql-odbc-09.06.0500-1.x86_64.rpm

- On Ubuntu and Debian systems, download the .deb file and then run the following command:
  
sudo gdebi <driver-installation-package>.deb

For example:
  
sudo gdebi tableau-postgresql-odbc_09.06.0501_amd64.deb

**Note:** The actual command will depend on the name of the driver installation package you download from the **Driver Download page**.

3. To validate that the drivers installed, navigate to the Administrative Views in Tableau Server.

**Initial Node Installation Defaults**

By default, the Tableau Server installer configures the number of process instances that Tableau Server runs based on the hardware detected by the installer. The default configuration applies to single-server installations and to the initial server of a multi-node installation.

You can calculate the default configuration based on the following rules for each process, where the number of cores refers to the number of logical CPU cores:

<table>
<thead>
<tr>
<th>Process Name</th>
<th>Number of Processes</th>
</tr>
</thead>
<tbody>
<tr>
<td>VizQL Server</td>
<td>Equal to the number of cores divided by four, up to a maximum of four process instances.</td>
</tr>
<tr>
<td>Backgrounder</td>
<td>Set to two unless the number of cores is fewer than eight.</td>
</tr>
<tr>
<td>Cache Server</td>
<td>Set to two unless the number of cores is fewer than eight.</td>
</tr>
<tr>
<td>Data Server</td>
<td>Set to two unless the number of cores is fewer than eight.</td>
</tr>
</tbody>
</table>
For all other process types, the number of process instances is set to one, regardless of the hardware.

Here's an example default configuration for a computer with 16-cores:

<table>
<thead>
<tr>
<th>Process Name</th>
<th>Number of Processes</th>
</tr>
</thead>
<tbody>
<tr>
<td>VizQL Server</td>
<td>4</td>
</tr>
<tr>
<td>Application Server</td>
<td>1</td>
</tr>
<tr>
<td>Backgrounder</td>
<td>2</td>
</tr>
<tr>
<td>Cache Server</td>
<td>2</td>
</tr>
<tr>
<td>Data Server</td>
<td>2</td>
</tr>
<tr>
<td>File Store</td>
<td>1</td>
</tr>
<tr>
<td>Data Engine</td>
<td>1</td>
</tr>
</tbody>
</table>

**Jump-start Installation**

This topic provides all of the steps required to perform a basic, quick-start installation of Tableau Server using the command line. The purpose of the configuration described here is to provide the quickest, simplest path to a Tableau Server installation on a computer running the CentOS 7.3 (and higher) or Ubuntu distribution of Linux. Use this procedure as practice, to try out the Tableau Server installation, management, and user experience before your actual server deployment.

**Important:** Do not use this procedure as a stand-alone resource for deploying Tableau Server into a production environment. To deploy Tableau Server into a production environment, refer to the content at Install and Configure Tableau Server.
The procedures in this topic will install an instance of Tableau Server for Linux with the following characteristics:

- Operating system: Supported RHEL-like Linux distribution, Ubuntu, or Debian.
- Identity store: local authentication
- Gateway port: 80
- Tableau Server administrator account: admin

Before you begin

Review the topic, Before you install.... The procedure below assumes that you have installed Linux on conforming hardware and according to the environmental requirements specified in that topic.

Note that the computer you install on must meet the minimum hardware requirements specified in Minimum Hardware Requirements and Recommendations for Tableau Server. The setup program will not install Tableau Server onto systems that do not meet the minimum hardware requirements.

Step 1: Install Tableau Server package and start Tableau Services Manager

Install Tableau Server with your distribution’s package manager, then run a script to initialize Tableau Services Manager (TSM). Tableau Services Manager is the management toolset used to install, configure, and manage Tableau services.

The initialize script is included with the installed package. For more details, see Install and Initialize TSM.

1. Log on as a user with sudo access to the computer where you want to install Tableau Server.
2. Navigate to the directory where you copied the Tableau Server installation package.
3. Use the package manager to install the Tableau Server package.
On RHEL-like distributions, including CentOS, run the following commands, where `<version>` is formatted as major-minor-maintenance (ex: `2019-2-5`):

```
sudo yum update
```
```
sudo yum install tableau-server-<version>.x86_64.rpm
```

**Note:** When you use `yum` to install Tableau Server, all dependent packages are automatically downloaded and installed. This is the preferred method for installing Tableau. If your organization does not allow you to use `yum` and you must install using `rpm -i`, you must also install all dependent packages separately.

On Ubuntu and Debian, run the following commands, where `<version>` is formatted as major-minor-maintenance (ex: `2019-2-5`):

```
sudo apt-get update
```
```
sudo apt-get upgrade
```
```
sudo apt-get -y install gdebi-core
```
```
sudo gdebi -n tableau-server-<version>_amd64.deb
```

4. Navigate to the `scripts` directory:

   ```
cd /opt/tableau/tableau_server/packages/scripts.<version_code>/
```

5. Run the following script to start TSM:

   ```
sudo ./initialize-tsm --accepteula
```

6. After initialization is complete, close the terminal session:

   ```
exit or logout
```
Step 2: Activate and register Tableau Server

Before you can configure Tableau Server you must activate a license and register. For more details, see Activate and Register Tableau Server.

1. Open a Bash session and sign in with the same account that you previously used to run `initialize-tsm`.
2. Activate the Tableau Server license. Run the following command:
   
   `tsm licenses activate -k <license_key>`

3. Register Tableau Server. Generate a template that you can edit by running the following command:

   `tsm register --template > /path/to/registration_file.json`

4. Open a text editing program, fill in the registration file, save it, then pass it with the following command:

   `tsm register --file /path/to/registration_file.json`

Step 3: Configure local identity store

You must configure the identity store settings. This procedure simplifies installation setting identity store to local authentication. Workbook examples are installed by default. For more details on customizing these defaults, see Configure Initial Node Settings.

- Pass the configuration file with the following command:

  `tsm settings import -f /opt/tableau/tableau_server-/packages/scripts.<version_code>/config.json`

Step 4: Finalize installation

The final steps of installation are to apply changes, initialize and start TSM, and then to create the administration account. More details about these steps are at Configure Initial Node
Settings.

1. Apply the configurations you made in the previous steps. Run the following command:

   ```bash
tsm pending-changes apply
   ```

2. Initialize and start Tableau Server. Run the following command:

   ```bash
tsm initialize --start-server --request-timeout 1800
   ```

3. Create the Tableau Server administrator account. Run the following command:

   ```bash
   tabcmd initialuser --server 'localhost:80' --username 'admin' --password '<password>'
   ```

   Where '='<password>' is a strong password. Enclose the password and other arguments in single quotes.

Use the admin account that you created to access the Tableau Server admin web pages. See Sign in to the Tableau Server Admin Area.

Step 5: Install PostgreSQL drivers

To validate that Tableau Server is installed and running properly and to review the built-in administrative views, you must install the PostgreSQL driver.

1. Download PostgreSQL drivers from the Driver Download page.

2. Install the drivers on your Tableau Server computer:

   - On CentOS and RHEL, download the .rpm file and then run the following command:

     ```bash
     sudo yum install <driver-installation-package>.rpm
     ```

     For example:

     ```bash
     sudo yum install tableau-postgresql-odbc-09.06.0500-1.x86_64.rpm
     ```
On Ubuntu and Debian systems, download the .deb file and then run the following command:

```
sudo gdebi <driver-installation-package>.deb
```

For example:

```
sudo gdebi tableau-postgresql-odbc_09.06.0501_amd64.deb
```

**Note:** The actual command will depend on the name of the driver installation package you download from the Driver Download page.

3. To validate that the drivers installed, navigate to the Administrative Views in Tableau Server.

### Configure Local Firewall

This topic explains how to configure the firewall on the computer running Tableau Server.

A local firewall should be enabled on the operating system to protect Tableau Server in single and multi-node deployments. In a distributed (multi-node) installation of Tableau Server, communication between nodes does not use secure communication. Therefore, you should enable firewalls on the computers that host Tableau Server.

We recommend that you configure the firewall so that only two ports are accessible to external traffic: the **gateway** port and the **tabadmincontroller** port. By default, these are ports 80 and 8850 respectively. Additionally, if you are running in a distributed deployment, then you will need to open the port range, 27000-27009, so licensing can communicate across nodes.

The **gateway** port is used for HTTP connection to Tableau Server. We recommend that you use SSL for the **gateway** port. If you will be using SSL, then the port must be 443 because Tableau Server does not support other ports for SSL. The procedures below describe how to configure the firewall for the **gateway** port. Configure the Tableau Server gateway (Configure Initial Node Settings) to match the port you set here.
The examples below describe how to configure the firewall on single and multi-node deployments of Tableau Server running on RHEL/CentOS distributions. The examples use Firewalld, which is the default firewall on CentOS.

Single-node configuration

1. Open a bash shell and run the following TSM command to retrieve the port number for the tabadmincontroller port:

   tsm topology list-ports

   Make a note of the tabadmincontroller port. By default, this port is 8850.

2. Start firewalld:

   sudo systemctl start firewalld

3. Verify that the default zone is a high-security zone, such as public. If it is not, we recommend changing it to a high-security zone.

   sudo firewall-cmd --get-default-zone

   sudo firewall-cmd --set-default-zone=public

4. Add ports for the gateway port and the tabadmincontroller port. In the example below we use the default ports (80 and 8850).

   sudo firewall-cmd --permanent --add-port=80/tcp

   sudo firewall-cmd --permanent --add-port=8850/tcp

5. Reload the firewall and verify the settings.

   sudo firewall-cmd --reload

   sudo firewall-cmd --list-all
Multi-node cluster configuration

In addition to enabling ports, configuring the firewall on a multi-node cluster requires additional steps to ensure that nodes can communicate with each other.

Before you begin

You will need the IP address for each node in the cluster. The example here uses <node1IP> as a placeholder for the initial node IP address, and <node2IP> and <node3IP> as placeholders for the IP addresses of two additional nodes.

Step 1: Configure initial node.

1. Open a bash shell and run the following TSM command to retrieve the port number for the tabadmincontroller port:

   tsm topology list-ports

   Make a note of the tabadmincontroller port. By default, this port is 8850.

2. Run the following commands to determine the range of port numbers that TSM may dynamically select. You will specify this range later in this procedure. Make a note of the port range.

   tsm configuration get -k ports.range.min

   tsm configuration get -k ports.range.max

   A typical range is 8000 to 9000.

3. Start firewall:

   sudo systemctl start firewalld

4. Verify that the default zone is a high-security zone, such as public. If it is not, we recommend changing it to a high-security zone.

   firewall-cmd --get-default-zone
Tableau Server on Linux Administrator Guide

```
sudo firewall-cmd --set-default-zone=public
```

5. **Add ports for the** gateway **port and the** tabadmincontroller **port. In the example below we use the default ports (80 and 8850). You must also add a port range (27000-27010) to enable licensing communication between nodes.**

```
sudo firewall-cmd --permanent --add-port=80/tcp
```

```
sudo firewall-cmd --permanent --add-port=8850/tcp
```

```
sudo firewall-cmd --permanent --add-port=27000-27010/tcp
```

6. **Configure the firewall to allow all traffic from the other nodes in the cluster. For the ports option, specify the range you noted in Step 2. Run the command for each of the additional nodes in your cluster. For example:**

```
sudo firewall-cmd --permanent --add-rich-rule='rule family=ipv4 source address=<node2IP>/32 port port=8000-9000 protocol=tcp accept'
```

```
sudo firewall-cmd --permanent --add-rich-rule='rule family=ipv4 source address=<node3IP>/32 port port=8000-9000 protocol=tcp accept'
```

7. **Reload the firewall and verify the settings.**

```
sudo firewall-cmd --reload
```

```
firewall-cmd --list-all
```

**Step 2: Configure additional nodes**

Each node in the cluster must be able to communicate with the initial node and with the other nodes.

Run this procedure on each additional node in the cluster. In this example, the node at IP address, `<node2IP>`, communicates with the initial node at `<node1IP>` and a third node at `<node3IP>`.
1. Start firewalld:

   ```
sudo systemctl start firewalld
   
   ```

2. Verify that the default zone is a high-security zone, such as public. If it is not, we recommend changing it to a high-security zone.

   ```
firewall-cmd --get-default-zone
   
sudo firewall-cmd --set-default-zone=public
   ```

3. Configure the firewall to allow gateway and tabadmincontroller access from the other nodes in the cluster. For example:

   ```
sudo firewall-cmd --permanent --add-rich-rule='rule family=ipv4 source address=<node1IP>/32 port port=80 protocol=tcp accept'

   sudo firewall-cmd --permanent --add-rich-rule='rule family=ipv4 source address=<node1IP>/32 port port=8000-9000 protocol=tcp accept'

   sudo firewall-cmd --permanent --add-rich-rule='rule family=ipv4 source address=<node3IP>/32 port port=80 protocol=tcp accept'

   sudo firewall-cmd --permanent --add-rich-rule='rule family=ipv4 source address=<node3IP>/32 port port=8000-9000 protocol=tcp accept'
   ```

   **In this example, since the tabadmincontroller port (8850) is included in the port range, it is not explicitly specified in a command.**

4. Reload the firewall and verify the settings.

   ```
sudo firewall-cmd --reload
   ```
Automated Installation of Tableau Server

Tableau provides an automated-installer script to automate an install of Tableau Server. The script is community supported. You can download the script and use it as written, or modify it for your specific needs.

Benefits of using the automated installer

- With a single command, you can install, configure and get to a working instance of Tableau Server.
- The command can be run without user input making it suitable for automation.
- The configuration can be set once and used for all your installations, making this a repeatable process.

When not to use the automated installer:

- If you are installing for the first time, we recommend that you manually test the installation before automating the process. Any issues that block installation are easier to resolve interactively, and after you have resolved these issues, you can use the automated installer.
- If you are testing or trying new configuration parameters such as authentication methods, we recommend that you manually run the installation first. TSM validates configuration entities and rejects configuration parameters that are not valid. Once you have the correct parameters identified, you can use the automated installer.
- If you are unable to or do not want to enter passwords into the secrets file, using the automated installer might not be an option for you.
Before you begin

Review the Before you install... topic to make sure you have installed Linux on a computer that meets the operating system requirements and the minimum hardware requirements for Tableau Server.

Note: If you are installing Tableau Server in a production environment, review the minimum hardware recommendations. The recommendations represent the minimum hardware configuration you should use for a production installation of Tableau Server.

To perform an automated installation, you have to use the automated installer package, which uses the Tableau Server install package as an input. We recommend that you download both of these packages before you begin as follows:

1. Download both the automated installer package and the Tableau Server installer package:
   a. Download the automated installer package from GitHub for the distribution you are using. The automated installer packages can be found in the packages sub directory.

   Note: The version of automated installer package you use must match the version of the Tableau Server installer package. For example, use 10.5.0 version of the automated installer package with 10.5.0 version of the Tableau Server installer package.

   b. Select and download the appropriate Tableau Server installer package from the Tableau Server Product Downloads and Release Notes page. The one you choose depends on which Linux distribution you are using. For example, for RHEL like systems, tableau-server-<version>.x86_64.rpm.

   c. Download the config.json, reg_templ.json, and the secrets templates.
2. Copy the packages and templates to a location on or accessible from the computer where you are going to install Tableau Server.

How to use the automated installer

The automated installer installs the Tableau Server installer package, creates the directories, sets the permissions required to run Tableau Server, and starts the Tableau Services Manager (TSM) setup. After the TSM setup is completed, the automated installer runs `tsm` commands to install, configure, and start Tableau Server. By default, during installation, the automated installer activates a trial license. If you have an actual product key, you can provide the product key at the command line or activate the product key after you run the script. Most of the command line options in the automated installer are the same as the options used by the `tsm initialize` command.

To run the automated installer without user input, you must provide the following required command line options:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-s &lt;secrets-file&gt;</code></td>
<td>The name of the secrets file. The secrets file should have the user names and passwords for TSM administrator and the Tableau Server administrator accounts.</td>
</tr>
</tbody>
</table>

**Note:** Providing the password in the secrets file is optional. However, if passwords are not found in the secrets file, you will be prompted to provide them during installation.

The automated installer package includes the template for the secrets file.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>-f &lt;config-file&gt;</code></td>
<td>The name of the configuration JSON file. The automated installer package includes the template for the configuration file.</td>
</tr>
<tr>
<td><code>-r &lt;registration-file&gt;</code></td>
<td>The name of the registration file. The automated installer package</td>
</tr>
<tr>
<td><strong>tration-file</strong></td>
<td>includes the template for the registration file.</td>
</tr>
<tr>
<td>----------------</td>
<td>-----------------------------------------------</td>
</tr>
<tr>
<td><strong>--accepteula</strong></td>
<td>Indicates that you have accepted the End User License Agreement.</td>
</tr>
<tr>
<td><strong>&lt;package-file&gt;</strong></td>
<td>The rpm or deb Tableau Server installer.</td>
</tr>
</tbody>
</table>

Use the **-h** option to see a full list of all the required command line options.

Configure Tableau Server for a forward proxy

If your organization uses a forward proxy solution to access the internet, then configure Tableau Server to use the proxy server. Tableau Server must access the internet for map data and for default licensing functionality.

We recommend configuring Tableau Server for a forward proxy solution during installation.

To configure proxy server during unattended installation, include the **--http_proxy** and/or **--https_proxy** flags to specify the forward proxy server.

Specify the URL with the port, for example:

```
--http_proxy=http://proxy.example.lan:80/ --https_proxy=http://1.2.3.4:443/
```

Take care to use **http** when you specify the URL for the **https_proxy** variable. Do not specify the **https** protocol for the value of the **https_proxy** variable.

To configure Tableau Server to bypass the forward proxy, include the **--no_proxy** flag. You should also add exceptions to this proxy configuration to guarantee that all communications within a local Tableau Server cluster (if you have one now or will have one later) do not route to the proxy server. For example:

```
--no_proxy=localhost,127.0.0.1,localaddress,.localdomain.com.
```

If you do not configure the forward proxy during installation, then refer to Configuring Tableau Server on Linux to work with a forward proxy, after you have installed.
Perform an unattended installation

Step 1: Install the automated installer

1. Log onto the computer as a user with sudo access.

2. Use the package manager to install the script package:

   - ON RHEL-like distributions, including CentOS, run the following command:

     ```
     sudo yum install /path/to/tableau-server-automated-installer-<version>.noarch.rpm
     ```

   - On Ubuntu and Debian, run the following commands:

     ```
     sudo apt-get update
     sudo apt-get -y install gdebi-core
     sudo gdebi -n /path/to/tableau-server-automated-installer-<version>.deb
     ```

   The automated installer package download includes templates for the configuration file (config.json), registration file (reg_templ.json) and the secrets (secrets) file that you can use to modify for your requirements as described in the next step. The installer script, and the templates for the initial node configuration, Tableau Server registration, and secrets file are installed to:

   ```
   /opt/tableau/tableau_server_automated_installer/automated-installer.<version>
   ```

Step 2: Create files to provide additional configuration information required to run the automated install

Since the automated installer is meant to run without user interaction, you must provide the following additional information:
1. Run the following command to copy the templates, config.json, reg_templ.json, and secrets, to another directory like your home directory. We don't recommend that you edit the template files directly:

   ```bash
   cp /opt/tableau/tableau_server_automated_installer-/automated-installer.<version>/{config.json,reg_templ.json,secrets} ~
   ```

2. Edit the configuration template, `config.json`, to provide the initial node configuration settings. You must provide identity store settings for the Tableau Server computer. Depending on your network requirements, you may need to also provide the gateway settings. The caching option is set to cache and reuse data for as long as possible. Sample workbooks are installed by default. The template includes the minimum required information, so the template is a starting point. For more information on configuration settings, see Configure Initial Node Settings.

3. Edit the registration file `reg_templ.json` to provide the information needed to register Tableau Server. For more information, see Activate and Register Tableau Server.

4. Edit the secrets file using the `secrets` template with the user name and password for the TSM administrator and Tableau Server administrator accounts.

   - The TSM administrator account should be the same user as the sudo admin running the script. If you do not want to specify the password in the secrets file, you can leave it blank, and you will be prompted to provide the password during installation.

   - The Tableau Server administrator account is the initial account that is created by the installer and is used to administer Tableau Server.

Step 3: Run the automated install

1. Log onto the computer as a user with sudo access.

   ON RHEL-like distributions, including CentOS, run the following command:
sudo /opt/tableau/tableau_server_automated_installer-automated-installer.<version>/automated-installer -s /path/to/secrets -f /path/to/config.json -r /path/to/reg_templ.json --accepteula /path/to/tableau-server-<version>_<version>_x86_64.rpm

On Ubuntu and Debian, run the following command:

sudo /opt/tableau/tableau_server_automated_installer-automated-installer.<version>/automated-installer -s /path/to/secrets -f /path/to/config.json -r /path/to/reg_templ.json --accepteula /path/to/tableau-server-<version>_<version>_amd64.deb

**Important:** You must specify `--accepteula` key to acknowledge and accept the end user license agreement (EULA) in the command that you use the run the script. The EULA is available in the following location: `/opt/tableau/tableau_server/packages/docs.<version_code>/EULA.rtf`

**Note:** If you are adding this machine as an additional node to an existing cluster, you must specify the `--bootstrap` flag and the node configuration file from the initial server. For more information on how to generate the node configuration file, see Install and Configure Additional Nodes
Install Tableau Server in a Disconnected (Air-Gapped) Environment

You can install Tableau Server in a disconnected environment that has no outside network access of any kind. Such disconnected environments, commonly referred to as air-gapped, are used when high security is needed to prevent data breaches or to guard against hacking. Air-gapped environments have no internet access, no outside network access, no outside wireless access, etc. The only means of getting software and data into or out of an air-gapped environment is by using removable media such as USB sticks or writeable optical CDs or DVDs.

Installing Tableau Server in an air-gapped environment is an advanced task for IT administrators who are familiar with the security considerations, best practices, and pitfalls of installing software in air-gapped environments.

The following Tableau Server features will be unavailable or will have reduced functionality in an air-gapped environment:

- **Maps** – Tableau Server uses externally hosted map data. Maps are unavailable in an air-gapped environment unless you also install a map server in your air-gapped environment. You can use the following workarounds.

- **Licensing** – Tableau Server needs to connect to the internet in order to activate product keys. However, you can manually activate the product keys.

- **External data** – Any data located outside your air-gapped environment is unavailable.

**Prerequisites**

In order to install Tableau Server in an air-gapped environment, you’ll need the following:
Trusted computer with limited access to the internet that you can use to download the installation packages and resources required by Tableau Server. A trusted computer has been scanned and cleared of any viruses and malware.

Trusted removable media that you can use to transfer the downloaded software to your air-gapped environment. Trusted removable media is removable media that is new and previously unused and comes from a reputable or known source. Trusted removable media has been scanned and verified that it does not contain any viruses or malware.

Air-gapped environment with computers and storage that meet the requirements for installing Tableau Server.

Installing Tableau Server on an Air-Gapped Computer Running Linux

The easiest way to install Tableau Server on a computer in an air-gapped environment is to do so before the computer is placed into the air-gapped environment. If that’s not possible you’ll need to download the required packages to a trusted computer outside the air gap:

1. On a trusted computer outside the air gap with internet access, download the Tableau Server installation package.

2. Extract the list of dependent packages:

   **On Ubuntu and Debian:**

   ```bash
dpkg --field <debfile> Depends
```

   Example command:

   ```bash
dpkg --field tableau-server-linux-1.deb Depends
```

   Example output:
ca-certificates, fontconfig, net-tools, bash-completion, ca-certificates-java, freeglut3, libegl1-mesa, libfreetype6, libgssapi-krb5-2, libxcomposite1, libxrender1, libxslt1.1, lsb-core

**On RHEL and RHEL-like Linux distributions:**

`yum -q deplist <RPM file>` (where `<RPM file>` is the .rpm package you downloaded from Tableau).

**Example command:**

`yum -q deplist tableau-server-linux_1.rpm`

**Example output:**

```
package: tableau-server-10400.17.0703.1600.x86_64 10400-17.0703.1600
dependency: /bin/sh
provider: bash.x86_64 4.2.46-21.el7_3
dependency: bash-completion
provider: bash-completion.noarch 1:2.1-6.el7
dependency: ca-certificates
provider: ca-certificates.noarch 2017.2.14-70.1.el7_3
dependency: fontconfig
provider: fontconfig.x86_64 2.10.95-10.el7
provider: fontconfig.i686 2.10.95-10.el7
dependency: freeglut
provider: freeglut.x86_64 2.8.1-3.el7
provider: freeglut.i686 2.8.1-3.el7
dependency: freetype
provider: freetype.x86_64 2.4.11-12.el7
provider: freetype.i686 2.4.11-12.el7
```
3. Download each of the dependent packages:

   **On Ubuntu and Debian:**

   ```bash
   apt-get download <package1> <package2>...
   ```

   **On RHEL and RHEL-like Linux distributions:**

   ```bash
   yumdownloader <package1> <package2>...
   ```

4. Transfer the packages to your removable media.

5. On your air-gapped computer, insert the removable media containing the Tableau...
Server installation package and dependent packages, and then run the installer.

6. After installation is complete, you can activate the Tableau Server product keys. For more information, see Activating Tableau Server in an Air-Gapped Environment.

Activating Tableau Server in an Air-Gapped Environment

Because an air-gapped computer is not connected to the internet, you’ll need to perform the Tableau Server activation process manually.

**Step 1. Transcribe data from your air-gapped system into an activation request template.**

1. On your Tableau Server in the air-gapped environment, use TSM to obtain the offline activation file. At a command prompt:

   ```
   tsm licenses get-offline-activation-file -k <product-key> -o <target-directory>
   ```

   The `<target-directory>` must exist. You can obtain your product key in the Tableau Customer Portal.

2. On a trusted computer that is connected to the internet and is capable of sending files via email, right-click and choose **Save link as** to download either the `server_firstpass_linux.tlq` or the `server_secondpass_linux.tlq` file. The firstpass template is used when the computer has never had Tableau Server installed on it before.

3. Open both the `offline.tlq` file from the disconnected computer, and the appropriate `server_firstpass_linux.tlq` or `server_secondpass_linux.tlq` file in an XML text editor such as Notepad++ on the trusted computer that has Internet access. You'll need to write down the values from the disconnected computer in order to copy them to the computer that has Internet access.

4. Update the following XML elements in the appropriate `.tlq` file (server_
firstpass_linux.tlq or server_secondpass_linux.tlq) with the values for the same elements in offline.tlq on the disconnected computer. The template file has X’s where you need to change the values. Do not change anything other than the values below.

All the Machine / Hash values in the .tlq files are Hex values. The only valid characters are 0 - 9 and A - F. Use all caps for any letters.

Line 2 - <EntitlementId>

Line 5 - <ClientVersion>

Line 5 - <RevisionType> (This value is only present in the server_secondpass_linux.tlq file.)

Line 5 - <MachineIdentifier> (This value is only present in the server_secondpass_linux.tlq file.)

Line 11 - <Value> (If the value is not present, remove the "X" place holder, leaving <Value></Value>.)

Line 12 - <Value> (If the value is not present, remove the "X" place holder, leaving <Value></Value>.)

Line 13 - <Value> (If the value is not present, remove the "X" place holder, leaving <Value></Value>.)

Line 15 - <SequenceNumber>

Line 61 - <Hash>

5. Upload the edited template (offline.tlq) to the Tableau Offline Activation website.

6. The website should say The activation was successful. Please click here to download your activation file.
Download the `activation.tlf` file and transfer it to your Tableau Server.

Step 2. Initialize or activate your license

1. Move the activation file (`activation.tlf`) to your disconnected computer using trusted removable media.

2. Run the following command:

   ```bash
   tsm licenses activate -f <path-and-activation-file>
   ```

First time license activation

If this is the first Tableau license activated on the computer, you will see this message:

Your license has been initialized. To complete the activation, we need one more exchange. Generate and send to Tableau a second activation request file.

You need to follow the steps below to activate your license. If you are not sure whether you need to repeat the steps, you can run this command:

```bash
 tsm licenses list
```

If you see a message like this one you need to repeat the steps above:

No licenses are currently activated.

Subsequent license activation

If this is not the first Tableau license activated on the computer, or if you have completed the steps above two times, you should see the message "Activation successful.,” which indicates that Tableau Server is activated.
**Note:** You may have to perform offline activation twice to completely activate your air-gapped Tableau Server. The first offline activation sets up Trusted Storage on the computer if Tableau has never been licensed on it before. The second offline activation propagates the licensing information into Trusted Storage. If you’re upgrading an existing install of Tableau Server, right-click and choose **Save link as** to download the `server_secondpass_linux.tiq` file. It will match all the places to put the values from the generated .tlq file.

### Displaying Maps in an Air-Gapped Environment

In an air-gapped environment, maps in Tableau Server will be unavailable due to the lack of internet access in the air-gapped environment. You can use the following workarounds to display maps in an offline environment:

- **Option 1** – Use the offline background maps

  To use the offline background map select **Map > Background maps > Offline**. For more information, see **Select Background Maps**.

  **Note:** Due to size limitations with the offline map, zooming in too deep may result in errors as the images for the more close-up maps are not stored offline.

- **Option 2** – Use a map as a background image

  Locate the desired map online and save it as an image file, or copy a map from another location. Set this map image as a static background image in Tableau Desktop. For more information, see **Use Background Images in Your Views**.

  **Note:** Zooming in and out will not work with background images.

- **Option 3** – Use a local GeoServer

  For more information, see **Use Web Map Service (WMS) Servers**.
Install Tableau Server in the Cloud

Even if you don’t have your own infrastructure and server hardware, you can deploy an enterprise-level Tableau Server installation in the cloud. Building a cloud-based solution has many benefits over an on-premises installation. For example, the overall total cost of ownership for building a Tableau Server solution in the cloud is normally much less than a similar on-premises solution because you don’t have to buy all of the expensive hardware. In addition, the cloud can provide better uptime, reliability, and fault-tolerance, especially if you deploy your solution across different regions and availability zones.


You can build and scale your Tableau environment in the following cloud environments:

- **Tableau Online** - If you don’t want to install software or purchase hardware, you can choose analytics as a service with Tableau Online, the Tableau-hosted environment in the cloud. For more information, see Tableau Online.

- **Amazon Web Services** - You install and manage Tableau Server on Amazon Web Services (AWS). For more information, see Install Tableau Server in the AWS Cloud.

- **Google Cloud Platform** - You install and manage Tableau Server on the Google Cloud Platform. For more information, see Install Tableau Server on the Google Cloud Platform.

- **Microsoft Azure** - You install and manage Tableau Server on Microsoft Azure. For more information, see Install Tableau Server on Microsoft Azure.

- **Alibaba Cloud** - You install and manage Tableau Server on Alibaba Cloud. For more information, see Install Tableau Server in the Alibaba Cloud.
Post Installation Tasks

After you install Tableau Server, you can perform other configuration tasks, such as reviewing the security hardening checklist, configuring server event notifications, configuring the data cache for views, and configuring server crash reporting.

Security Hardening Checklist

The following list provides recommendations for improving the security ("hardening") of your Tableau Server installation.

1. Update to the current version

We recommend that you always run the latest version of Tableau Server. Additionally, Tableau periodically publishes maintenance releases of Tableau Server that include fixes for known security vulnerabilities. (Information regarding known security vulnerabilities can be found on the Security Bulletins page.) We recommend that you review maintenance release notifications to determine whether you should install them.

To get the latest version or maintenance release of Tableau Server, visit the Customer Portal page.

2. Configure SSL/TLS with a valid, trusted certificate

Secure Sockets Layer (SSL/TLS) is essential for helping to protect the security of communications with Tableau Server. Configure Tableau Server with a valid, trusted certificate (not a self-signed certificate) so that Tableau Desktop, mobile devices, and web clients can connect to the server over a secured connection. For more information, see SSL.

3. Disable older versions of TLS

Tableau Server uses TLS to authenticate and encrypt many connections between components and with external clients. External clients, such as browsers, Tableau Desktop, Tableau Mobile connect to Tableau using TLS over HTTPS. Transport layer security (TLS)
is an improved version of SSL. In fact, older versions of SSL (SSL v2 and SSL v3) are no longer considered to be adequately secure communication standards. As a result, Tableau Server does not allow external clients to use SSL v2 or SSL v3 protocols to connect. We recommend that you only allow external clients to connect to Tableau Server with TLS v1.2.

Specifically, we recommend that you disable TLS v1 and TLS v1.1 on Tableau Server. However, before you disable a specific version of TLS, verify that the browsers that your users connect to Tableau Server with support TLS v1.2. In some cases, you may need to preserve support for TLSv1.1.

The following tsm command enables TLS v1.2 (using the "all" parameter) and disables SSL v2, SSL v3, TLS v1, and TLS v1.1 (by prepending the minus [-] character to a given protocol).

```
tsm configuration set -k ssl.protocols -v 'all -SSLv2 -SSLv3 -TLSv1 -TLSv1.1'
```

```
tsm pending-changes apply
```

The `pending-changes apply` command displays a prompt to let you know this will restart Tableau Server if the server is running. The prompt displays even if the server is stopped, but in that case there is no restart. You can suppress the prompt using the `--ignore-prompt` option, but this does not change the restart behavior. For more information, see tsm pending-changes apply.

4. Configure SSL encryption for internal traffic

Configure Tableau Server to use SSL to encrypt all traffic between the Postgres repository and other server components. By default, SSL is disabled for communications between server components and the repository. We recommend enabling internal SSL for all instances of Tableau Server, even single-server installations. Enabling internal SSL is especially important for multi-node deployments. See Configure SSL for Internal Postgres Communication.
5. Enable firewall protection

Tableau Server was designed to operate inside a protected internal network.

**Important:** Do not run Tableau Server, or any components of Tableau Server on the internet or in a DMZ. Tableau Server must be run within the corporate network protected by an internet firewall. We recommend configuring a reverse proxy solution for internet clients that need to connect to Tableau Server. See Configuring Proxies for Tableau Server.

A local firewall should be enabled on the operating system to protect Tableau Server in single and multi-node deployments. In a distributed (multi-node) installation of Tableau Server, communication between nodes does not use secure communication. Therefore, you should enable firewalls on the computers that host Tableau Server. See Configure Local Firewall.

To prevent a passive attacker from observing communications between nodes, configure a segregated virtual LAN or other network layer security solution.

See Tableau Services Manager Ports to understand which ports and services Tableau Server requires.

6. Restrict access to the server computer and to important directories

Tableau Server configuration files and log files can contain information that is valuable to an attacker. Therefore, restrict physical access to the machine that is running Tableau Server. In addition, make sure that only authorized and trusted users have access to the Tableau Server files in the `/var/opt/tableau/tableau_server/` directory.

7. Generate fresh secrets and tokens

Any Tableau Server service that communicates with repository or the cache server must first authenticate with a secret token. The secret token is generated during Tableau Server
setup. The encryption key that internal SSL uses to encrypt traffic to Postgres repository is also generated at during setup.

We recommend that after you install Tableau Server, you generate new encryption keys for your deployment.

These security assets can be regenerated with the `tsm security regenerate-internal-tokens` command.

Run the following commands:

```bash
tsm security regenerate-internal-tokens

tsm pending-changes apply
```

8. Disable services that you're not using

To minimize the attack surface of the Tableau Server, disable any connection points that are not needed.

REST API

The REST API interface is enabled by default. If no applications will make REST API calls to your installation of Tableau Server 9.3 (or later), disable it by using the following commands:

```bash
tsm configuration set -k api.server.enabled -v false

tsm pending-changes apply
```

**Important:** Tableau Prep uses REST API to access Tableau Server. If your organization uses Tableau Prep, do not disable REST API.
JMX Service

JMX is disabled by default. If it’s enabled but you’re not using it, you should disable it by using the following:

```bash
tsm configuration set -k service.jmx_enabled -v false
tsm pending-changes apply
```

9. Verify session lifetime configuration

By default, Tableau Server does not have an absolute session timeout. This means that client sessions can remain open indefinitely if the Tableau Server inactivity timeout is not exceeded. (The default inactivity timeout is 240 minutes.)

If your security policy requires it, you can set an absolute session timeout. Be sure to set your absolute session timeout in a range that allows the longest-running extract or publishing operations in your organization. Setting the session timeout too low may result in extract and publishing failures for long-running operations.

To set the session timeout run the following commands:

```bash
tsm configuration set -k wgserver.session.apply_lifetime_limit -v true
tsm configuration set -k wgserver.session.lifetime_limit -v value, where value is the number of minutes. The default is 1440, which is 24 hours.
tsm configuration set -k wgserver.session.idle_limit -v value, where value is the number of minutes. The default is 240.
tsm pending-changes apply
```
10. Configure a server safelist for file-based data sources

By default, Tableau Server allows authorized Tableau Server users to build workbooks that use files on the server as file-based data sources (such as spreadsheets). In this scenario, files are accessed by the tableau system account.

To prevent unwanted access to files, we recommend that you configure safelist (sometimes referred to as "whitelist") functionality. This lets you limit tableau account access to just the directory paths where you host data files.

1. On the computer running Tableau Server, identify the directories where you will host data source files.

   **Important** Make sure the file paths you specify in this procedure exist on the server. If the paths do not exist when the computer starts, Tableau Server will not start.

2. Run the following commands:

   ```
   tsm configuration set -k native_api.allowed_paths -v "path", where path is the directory to add to the safelist. All subdirectories of the specified path will be added to the safelist. If you want to specify multiple paths, separate them with a semicolon, as in this example:
   
   tsm configuration set -k native_api.allowed_paths -v 
   "/datasources;/HR/data"
   
   tsm pending-changes apply
   ```

11. Enable HTTP Strict Transport Security for web browser clients

HTTP Strict Transport Security (HSTS) is a policy configured on web application services, such as Tableau Server. When a conforming browser encounters a web application run-
Tableau Server on Linux Administrator Guide

Enabling HSTS, then all communications with the service must be over a secured (HTTPS) connection. HSTS is supported by major browsers.

For more information about how HSTS works and the browsers that support it, see The Open Web Application Security Project web page, HTTP Strict Transport Security Cheat Sheet.

To enable HSTS, run the following commands on Tableau Server:

```
 tsm configuration set -k gateway.http.hsts -v true
```

By default, HSTS policy is set for one year (31536000 seconds). This time period specifies the amount of time in which the browser will access the server over HTTPS. You should consider setting a short max-age during initial roll-out of HSTS. To change this time period, run:

```
 tsm configuration set -k gateway.http.hsts_options -v max-age=<seconds>
```

For example, to set HSTS policy time period to 30 days, enter:

```
 tsm configuration set -k gateway.http.hsts_options -v max-age=2592000.
```

```
 tsm pending-changes apply
```

**12. Disable Guest access**

Core-based licenses of Tableau Server include a Guest user option, which allows any user in your organization to see and interact with Tableau views embedded in web pages.

Guest user access is enabled by default on Tableau Servers deployed with core-based licensing.

Guest access allows users to see embedded views. The Guest user cannot browse the Tableau Server interface or see server interface elements in the view, such as user name, account settings, comments, and so on.

If your organization has deployed Tableau Server with core licensing and Guest access is not required, then disable Guest access.

You can disable Guest access at the server or site level.
You must be a server administrator to disable the Guest account at either the server or the site level.

**To disable Guest access at the server level:**

1. In the site menu, click **Manage All Sites** and then click **Settings > General**.
2. For **Guest Access**, clear the **Enable Guest account** check box.
3. Click **Save**.

**To disable Guest access for a site:**

1. In the site menu, select a site.
2. Click **Settings**, and on the Settings page, clear the **Enable Guest account** check box.

For more information, see Guest User.

**13. Set referrer-policy HTTP header to 'same-origin'**

Beginning in 2019.2, Tableau Server includes the ability to configure Referrer-Policy HTTP header behavior. This policy is enabled with a default behavior that will include the origin URL for all "secure as" connections (**no-referrer-when-downgrade**), which sends origin referrer information only to like connections (HTTP to HTTP) or those that are more secure (HTTP to HTTPS).

However, we recommend setting this value to **same-origin**, which only sends referrer information to same-site origins. Requests from outside the site will not receive referrer information.

To update the referrer-policy to **same-origin**, run the following commands:

```bash
tsm configuration set -k gateway.http.referrer_policy -v same-origin
tsm pending-changes apply
```
For more information about configuring additional headers to improve security, see HTTP Response Headers.

Change List

<table>
<thead>
<tr>
<th>Date</th>
<th>Change</th>
</tr>
</thead>
<tbody>
<tr>
<td>September 2017</td>
<td>Ported and updated for Tableau Services Manager and Linux platform.</td>
</tr>
<tr>
<td>May 2018</td>
<td>Added clarification: Do not disable REST API in organizations that are running Tableau Prep.</td>
</tr>
<tr>
<td>May 2019</td>
<td>Added recommendation for referrer-policy HTTP header.</td>
</tr>
<tr>
<td>June 2019</td>
<td>Removed recommendation to disable Triple-DES. As of version 2019.3, Triple-DES is no longer a default supported cipher for SSL. See What’s Changed - Things to Know Before You Upgrade.</td>
</tr>
</tbody>
</table>

Configure SMTP Setup

Tableau Server can email server administrators about system failures, and email server users about subscribed views and data-driven alerts. First, however, you need to configure the SMTP server that Tableau Server uses to send email. After configuring SMTP, complete the steps to configure notifications (Configure Server Event Notification), then when you start or restart the server, it will trigger an email notification, which confirms that you have set up notifications correctly.

Configuring SMTP requires that you restart Tableau Server services.

Secure SMTP

To enable and configure TLS for SMTP, you must use the TSM CLI as described in this topic.
If your organization does not use public certificates for verifying TLS connections, then you can upload a private certificate to Tableau Server to verify trusted connections. For more information, see the tsm security custom-cert add command.

You may also configure SMTP TLS for encryption only by disabling the certificate validation process. See the section, Configuration file reference, later in this topic for more information.

Use the TSM web interface

1. Open TSM in a browser:

   https://<tsm-computer-name>:8850. For more information, see Sign in to Tableau Services Manager Web UI.

2. Click **Notifications** on the **Configuration** tab and click **Email Server**.

3. Enter the SMTP configuration information for your organization:

   ![Email Server Configuration](image)

   - **SMTP server address**: smtp.example.lan
   - **Username**: tableau.notify@example.lan
   - **Password**: **************
   - **Port Number**: 25 (Default)
   - **Send all emails from**: no-reply@example.lan
   - **Send server health email to**: tableau.health@example.lan
   - **Tableau Server URL**: https://tableau.example.lan

4. Click **Save Pending Changes** after you’ve entered your configuration information.

5. Click **Pending Changes** at the top of the page:
6. Click **Apply Changes and Restart**.

7. Run the `tsm email test-smtp-connection` to view and verify the connection configuration. See `tsm email test-smtp-connection`.

**Use the TSM CLI**

For the initial configuration of SMTP, we recommend that you use the configuration file template below to create a json file. You can also set any single configuration key listed below with the syntax described in `tsm configuration set`.

1. Copy the following json template to a file.

```json
{
  "configKeys": {
    "svcmonitor.notification.smtp.server": "SMTP server host name",
    "svcmonitor.notification.smtp.send_account": "SMTP user name",
    "svcmonitor.notification.smtp.port": 443,
  }
}
```

**Important**: The template below include common options for most deployments. After you copy the template to a text file, you must edit the option values for your SMTP server requirements. You may need to remove or add options. See the reference section that follows for more information about all supported SMTP key options.
"svcmonitor.notification.smtp.password": "SMTP server password",
"svcmonitor.notification.smtp.ssl_enabled": true,
"svcmonitor.notification.smtp.from_address": "From email address",
"svcmonitor.notification.smtp.target_addresses": "To email address1,address2",
"svcmonitor.notification.smtp.canonical_url": "Tableau Server URL"
}
}

2. Run the `tsm settings import -f file.json` to pass the json file with the appropriate values to Tableau Services Manager to configure Tableau Server for SMTP. Tableau Services Manager will validate the entity values.

3. Run the `tsm pending-changes apply` command to apply the changes. See `tsm pending-changes apply`.

4. Run the `tsm email test-smtp-connection` to view and verify the connection configuration. See `tsm email test-smtp-connection`.

Configuration file reference

This table lists all of the options that can be used to configure SMTP.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>svc-monitor.notification.smtp.server</code></td>
<td>Address of SMTP server.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td>&quot;svc-monitor.notification.smtp.server&quot;</td>
<td></td>
</tr>
</tbody>
</table>

Tableau Server on Linux Administrator Guide
<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>itor.notification.smtp.server</code></td>
<td>User name for SMTP account.</td>
</tr>
<tr>
<td><code>monitor.notification.smtp.send_account</code></td>
<td>Port number for SMTP server. The default is 25.</td>
</tr>
<tr>
<td><code>monitor.notification.smtp.port</code></td>
<td>Password for SMTP server account.</td>
</tr>
<tr>
<td>Example:</td>
<td>Example:</td>
</tr>
<tr>
<td></td>
<td>&quot;monitor.notification.smtp.password&quot;:&quot;password&quot;</td>
</tr>
<tr>
<td><code>monitor.notification.smtp.ssl_enabled</code></td>
<td>Specifies whether the connection to the SMTP server is encrypted. The default is false.</td>
</tr>
<tr>
<td><code>monitor.notification.smtp.ssl_required</code></td>
<td>If enabled, Tableau Server will refuse to connect to SMTP servers without using TLS. The <code>monitor.notification.smtp.ssl_enabled</code> option must also be set to true. The default is false.</td>
</tr>
<tr>
<td><code>monitor.notification.smtp.ssl_required</code></td>
<td>If set to true, Tableau Server will check the SMTP server identity as specified by RFC 2595. These additional checks based on the content of</td>
</tr>
<tr>
<td>Option</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------------------------</td>
<td>-----------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>check_server_identity</td>
<td>the server's certificate are intended to prevent man-in-the-middle attacks. The default is false.</td>
</tr>
<tr>
<td>svc-monitor.notification.smtp.ssl_trust_all_hosts</td>
<td>When using TLS, trust certificates from all mail servers, ignoring the validity of the certificate's chain of trust. By setting this key to true, TLS will be used only to encrypt the traffic to the SMTP host. The default is false.</td>
</tr>
<tr>
<td>svc-monitor.notification.smtp.ssl_ciphers</td>
<td>The default and supported sets of cipher suites is defined by the version of JDK that is installed with Tableau Server. See the section below, TLS ciphers, for a list of supported and default ciphers. To update the cipher suites used by Tableau Server for SMTP TLS connections, enter a white space-separated list of cipher suites for this value. For example, &quot;TLS_ECDH_ECDSA_WITH_AES_128_CBC_SHA256 TLS_DHE_DSS_WITH_AES_128_GCM_SHA256 TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384&quot;.</td>
</tr>
<tr>
<td>svc-monitor.notification.smtp.ssl_versions</td>
<td>The default TLS versions enabled on this version of Tableau Server are TLSv1, TLSv1.1, TLSv1.2. TLS version support is defined by the version of</td>
</tr>
<tr>
<td>Option</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------------------------</td>
<td>-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>JDK that is installed with Tableau Server</strong></td>
<td><strong>Supported versions of TLS are</strong> SSLv2Hello, SSLv3, TLSv1, TLSv1.1, TLSv1.2.</td>
</tr>
<tr>
<td><strong>To update the versions used by Tableau Server for SMTP TLS connections, enter a white space-separated list of versions for this value.</strong> For example, &quot;TLSv1.1 TLSv1.2&quot;.</td>
<td></td>
</tr>
<tr>
<td><strong>svc-monitor.notification.smtp.from_address</strong></td>
<td>Email address that will send an notification if there's a system failure. The email address must have valid syntax (for example, <a href="mailto:ITalerts@bigco-.com">ITalerts@bigco-.com</a> or noreply@mycompany), but it does not have to be an actual email account on Tableau Server. (Some SMTP servers may require an actual email account, however.) Note: You can override the system-wide email address on a per-site basis. For more information, see What is a site. Example: &quot;svc-monitor.notification.smtp.from_address&quot;: &quot;<a href="mailto:donot-reply@example.com">donot-reply@example.com</a>&quot;</td>
</tr>
<tr>
<td><strong>svc-monitor.notification.smtp.target_addresses</strong></td>
<td>Email address to receive notifications. If email notifications are enabled, you need to include at least one address. Separate multiple addresses with commas.</td>
</tr>
<tr>
<td>Option</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>svc-monitor.notification.smtp.target_addresses</td>
<td>Example:</td>
</tr>
<tr>
<td></td>
<td>&quot;svc-monitor.notification.smtp.target_addresses&quot;: &quot;<a href="mailto:iluvdata@example.com">iluvdata@example.com</a>&quot;</td>
</tr>
<tr>
<td>svc-monitor.notification.smtp.canonical_url</td>
<td>URL of the Tableau Server. Enter http:// or https://, followed by the name or IP address of the Tableau server. Used in the footer of subscription email.</td>
</tr>
<tr>
<td></td>
<td>Example:</td>
</tr>
<tr>
<td></td>
<td>&quot;svc-monitor.notification.smtp.canonical_url&quot;: &quot;<a href="http://myserver.example.com">http://myserver.example.com</a>&quot;</td>
</tr>
</tbody>
</table>

**TLS ciphers**

The following is a list of TLS ciphers that are supported by the JDK that is included with Tableau Server 2019.4. In this version of Tableau Server, all of these ciphers are enabled by default. You can specify a custom list of ciphers for your SMTP configuration by entering a white-space separated list with the option, `svc-monitor.notification.smtp.ssl_ciphers`, as described in the table above.

<table>
<thead>
<tr>
<th>TLS_RSA_WITH_AES_128_CBC_SHA256</th>
<th>TLS_ECDH_RSA_WITH_AES_256_GCM_SHA384</th>
</tr>
</thead>
<tbody>
<tr>
<td>TLS_DHE_DSS_WITH_AES_256_GCM_SHA384</td>
<td>TLS_ECDH_RSA_WITH_AES_128_GCM_SHA256</td>
</tr>
<tr>
<td>TLS_ECDH_RSA_WITH_AES_256_CBC_SHA384</td>
<td>TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA</td>
</tr>
<tr>
<td>-----------------------------------</td>
<td>--------------------------------------</td>
</tr>
<tr>
<td>TLS_ECDH_ECDSA_WITH_AES_128_CBC_SHA</td>
<td>TLS_RSA_WITH_AES_256_CBC_SHA256</td>
</tr>
<tr>
<td>TLS_RSA_WITH_AES_128_GCM_SHA256</td>
<td>TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384</td>
</tr>
<tr>
<td>TLS_ECDH_RSA_WITH_AES_128_CBC_SHA256</td>
<td>TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256</td>
</tr>
<tr>
<td>TLS_DHE_DSS_WITH_AES_128_CBC_SHA256</td>
<td>TLS_DHE_DSS_WITH_AES_256_CBC_SHA</td>
</tr>
<tr>
<td>TLS_DHE_RSA_WITH_AES_128_CBC_SHA256</td>
<td>TLS_DHE_RSA_WITH_AES_256_CBC_SHA256</td>
</tr>
<tr>
<td>TLS_RSA_WITH_AES_256_GCM_SHA384</td>
<td>TLS_EMPTY_RENEGOTIATION_INFO_SCSV</td>
</tr>
<tr>
<td>TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384</td>
<td>TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA</td>
</tr>
<tr>
<td>TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA</td>
<td>TLS_DHE_DSS_WITH_AES_256_CBC_SHA256</td>
</tr>
<tr>
<td>TLS_RSA_WITH_AES_256_CBC_SHA</td>
<td>TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA256</td>
</tr>
<tr>
<td>TLS_ECDH_ECDSA_WITH_AES_128_CBC_SHA256</td>
<td>TLS_DHE_RSA_WITH_AES_256_CBC_SHA</td>
</tr>
<tr>
<td>TLS_DHE_DSS_WITH_AES_128_GCM_SHA256</td>
<td>TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256</td>
</tr>
<tr>
<td>TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA</td>
<td>TLS_ECDH_RSA_WITH_AES_128_CBC_SHA</td>
</tr>
</tbody>
</table>

Tableau Server on Linux Administrator Guide
Files and Permissions in TSM

This topic covers the permissions requirements needed for Tableau Services Manager (TSM) to access and use files. This information is intended for server administrators. This topic does not address permissions used for managing users and content within Tableau Server (permissions for content and users). For information related to these permissions, see Permissions.

During installation of TSM and Tableau Server, an unprivileged user (tableau) is created and added to a server authorized group (tableau). This user account enables the work done
Tableau Server on Linux Administrator Guide

by TSM and Tableau Server processes. You can change the user and group during installation. For more information, see Identity Store.

Permissions requirements for TSM apply to both files, and to the directories in which the files are placed. When TSM creates and manages files, the files get put into specific default locations with the necessary permissions and you don’t need to worry about setting permissions. When you create, copy, or move files yourself, or when you put files into non-default locations, you need to be aware of permission requirements so that TSM can properly access the files. Common cases (For information about using non-default locations, see tsm File Paths.)

General rules for permissions and TSM are:

- Files—if the `tableau` group has access to a file (if it is the group owner and has read access to the file), the users in the group have access to the file. An alternate approach is to give "other" read access.

- Directories—if the `tableau` group has read and execute access to the directory that contains a file, and any parent directories of that directory, the users in the group have access to the file.

Situations that may require you to adjust permissions include server backup files and site import archives that you copy from a different computer or to a non-default location, customization files such as logos or images, and security certificates such as SSL certificates.

For example, if you migrate from Tableau Server on Windows to Tableau Server on Linux, you use a backup created in Windows to restore data to your Linux server. Because this backup file isn't created by TSM, it may not have the correct permissions for the restore process to access it. You need to make sure the backup file and the directory structure you copy it into have the proper permissions. Similarly, if you are copying files like certificates to additional nodes in a cluster, you need to make sure the files and the directories you copy them into have the permissions the `tableau` user needs in order to access them.
Setting permissions for individual files

If you are using a file you copy to one of the default locations created by TSM, you need to make sure the ownership and permissions on the file allow TSM access by giving the tableau user read access. You can do this in one of two ways:

- You can give the tableau user read access by giving the tableau group (in a default installation) read and execute access to a file using the chgrp and chmod commands. For example:

  chgrp tableau <backup>.tsbak
  chmod g+rx <backup>.tsbak

- Alternately, you can give world read and execute access to the file:

  chmod o+rx <backup>.tsbak

Setting permissions for directories

In addition to setting the proper permissions on the files themselves, TSM also needs permissions for the directory that contains the file, as well as any parent directories. If you are using a non-default location for files that TSM will access, you will need to make sure permissions for the parent directory or directories that contain the file allow read and execute access.

You can address this issue in a couple of ways:

- Change group ownership of the directory to the tableau group, and add group read and execute permission to the directory. Doing this makes files in the directory more available to the tableau user.

  chgrp tableau <directory-name>
  chmod g+rx <directory-name>

- Alternatively, you can add world read and execute permission to the directory. This
makes files in the directory more available to all users on the system. This approach may require additional steps to ensure security of other files in the directory. For example, you may want to make sure other files in the directory are not world readable so other users cannot read them.

```
chmod o+rx <directory-name>
```

**Hint:** You can use `namei -mo` command to list an entire permissions tree. This can make it easier to see what directories need to have permissions adjusted to allow access by the *tableau* group. You can find more information on the internet.

---

**Configure Server Event Notification**

A Tableau Services Manager (TSM) administrator can configure Tableau Server to allow notifications for the following events:

- Content updates
  - Extract failures (enabled by default)
  - Subscription views for users (disabled by default)
- Server health monitoring
  - Server status changes (disabled by default)
  - License reporting (disabled by default)
- Drive space
  - Email alerts when space crosses or remains below pre-configured thresholds (disabled by default)
  - Recording usage history (enabled by default)

**Note:** You need to configure SMTP before you can configure subscriptions or notifications. For more information, see Configure SMTP Setup.

Use the TSM web interface
1. Open TSM in a browser:

https://<tsm-computer-name>:8850.

For more information, see Sign in to Tableau Services Manager Web UI.

2. Click **Notifications** on the **Configuration** tab and click **Events**.

3. Configure notification settings for your organization:
   - Content updates
     - **Send emails for extract refresh failures**
       When this option is enabled (the default), a server administrator can configure email notifications to be sent when extract refreshes fail. These messages are configured at the site level, so even if this option is enabled, messages are not sent unless the **Send email to data source and workbook owners when scheduled refreshes fail** option is enabled for a site (this is enabled by default). For details, see Enable Extract Refresh Scheduling and Failure Notification.

     - **Allow users to receive email for views that they have subscribed to**
       When this option is enabled (by default is it disabled), a server administrator can configure a site to send subscription email. These email messages are configured at the site level and can only be configured when this option is enabled. For details, see Set Up a Site for Subscriptions.

       When users subscribe to a workbook or view, a snapshot of the view is emailed to them on a scheduled basis, so they can see the latest updates without having to sign into Tableau Server.
To allow users to attach PDF renderings on subscription emails, select

**Let users add attachments to subscribed views.**

- Server health monitoring

- **Send emails for Tableau Server process events (up, down, and failover)**

  Tableau Server sends an email message when the data engine, repository, or gateway server processes stop or restart, or when the initial Tableau Server node stops or restarts.

  If you are running a single-server installation (all processes on the same computer), health alerts are only sent when Tableau Server is up. No "down" alerts are sent. If you are running a distributed installation that's configured for failover, a DOWN alert means that the active repository or a data engine instance has failed and the subsequent UP alert means that the passive instance (repository) or second instance (data engine) of that process has taken over.

- **Send emails for Tableau Server license reporting**

  License reporting is generated at Tableau Desktop and sent to Tableau Server.

- **Drive space**

  Enable notifications for remaining drive space on your Tableau Server.

  - **Send emails when unused drive space drops below thresholds**

    You can configure Tableau Server to send email alerts when disk space usage on any node crosses a threshold, or remains below the threshold. And you can configure how often threshold alerts are sent.
There are two thresholds you must set, **Warning threshold** and **Critical threshold**. Thresholds are expressed in percentage of disk space remaining. The critical threshold must be less than the warning threshold.

You also specify the **Send threshold alert every** option. This determines how often, in minutes, warning and critical alerts should be sent. The default value is 60 minutes.

- **Record disk space usage information and threshold violations for use in custom administrative views**

When you configure Tableau Server to record disk space usage, information about free disk space is saved in the repository and you can view the usage history using the Administrative Views.

4. Click **Save Pending Changes** after you’ve entered your configuration information.

5. Click **Pending Changes** at the top of the page:

   ![Pending Changes](image)

6. Click **Apply Changes and Restart**.

**Use the TSM CLI**

The various notification values described above can be set individually with the tsm configuration set command. Alternatively, you can construct a json file and pass all configuration values in one operation. Both methods are described in this section.

Set notification values individually

The following table shows the key/value pairs that map to the notification events described earlier in this topic. Use the tsm configuration set command with the following syntax to set a single key/value pair:
### Tableau Server on Linux Administrator Guide

```bash
  tsm configuration set -k <config.key> -v <config_value>
```

For example, to enable job failure notifications, run the following command:

```bash
  tsm configuration set -k backgrounder.notifications_enabled -v true
```

<table>
<thead>
<tr>
<th>Notification option</th>
<th>Key</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Extract failures or Flow run failures</td>
<td><code>backgrounder.notifications_enabled</code></td>
<td>`true</td>
</tr>
<tr>
<td>Enable subscription views for user</td>
<td><code>subscriptions.enabled</code></td>
<td>`true</td>
</tr>
<tr>
<td>Enable PDF attachments for subscriptions</td>
<td><code>subscriptions.attachments_enabled</code></td>
<td>`true</td>
</tr>
<tr>
<td>Maximum attachment size (MB) for subscription notifications</td>
<td><code>subscriptions.max_attachment_size_megabytes</code></td>
<td><code>integer value, default is 150</code></td>
</tr>
<tr>
<td>Server status changes</td>
<td><code>svcmonitor.notification.smtp.enabled</code></td>
<td>`true</td>
</tr>
<tr>
<td>License reporting</td>
<td><code>features.DesktopReporting</code></td>
<td>`true</td>
</tr>
<tr>
<td>Remaining space thresholds: enable email</td>
<td><code>storage.monitoring.email_enabled</code></td>
<td>`true</td>
</tr>
</tbody>
</table>
## Notifications

<table>
<thead>
<tr>
<th>Remaining space thresholds: warning percentage</th>
<th>storage.monitoring.warning_percent</th>
<th>integer value, for example, 20</th>
</tr>
</thead>
<tbody>
<tr>
<td>Remaining space thresholds: critical percentage</td>
<td>storage.monitoring.critical_percent</td>
<td>integer value, for example, 15</td>
</tr>
<tr>
<td>Set email interval</td>
<td>storage.monitoring.email_interval_min</td>
<td>integer value, in minutes, for example, 25</td>
</tr>
<tr>
<td>Record usage history</td>
<td>storage.monitoring.record_history_enabled</td>
<td>true</td>
</tr>
</tbody>
</table>

After you are done setting values, you must run the following command:

```bash
tsm pending-changes apply
```

The `pending-changes apply` command displays a prompt to let you know this will restart Tableau Server if the server is running. The prompt displays even if the server is stopped, but in that case there is no restart. You can suppress the prompt using the `--ignore-prompt` option, but this does not change the restart behavior. For more information, see `tsm pending-changes apply`.

Set all notification values with a single json file

To make all notifications settings with a single configuration, you can pass a json file.

Copy and edit the following template to create a file for your configuration.

```json
{
    "configKeys": {
```
After you have saved the file, pass it with the following command:

```bash
csh tsm settings import -f <path-to-file.json>
```

To apply changes, run the following command:

```bash
csh tsm pending-changes apply
```

The `pending-changes apply` command displays a prompt to let you know this will restart Tableau Server if the server is running. The prompt displays even if the server is stopped, but in that case there is no restart. You can suppress the prompt using the `--ignore-prompt` option, but this does not change the restart behavior. For more information, see `tsm pending-changes apply`.

### Configure Data Cache

Views published to Tableau Server are interactive and sometimes have a live connection to a database. As users interact with the views in a web browser, the data that is queried gets stored in a cache. Subsequent visits will pull the data from this cache if it is available. By default, Tableau Server will cache and reuse data for as long as possible. You can change
this behavior by configuring the caching option using the `tsm data-access caching set` command.

1. Run this command:

   ```bash
tsm data-access caching set -r <value>
   ```

   Where `<value>` is one of these options:

   - **low or empty string (""").** This is the default value and indicates that Tableau Server should configure cache and always use cached data when available.
   - `<value>" specifies the maximum number of minutes data should be cached.
   - **always** or 0 (zero). These values indicates that Tableau Server should always get the latest data and that the cache should be refreshed each time a page is reloaded.

2. Apply changes with the `tsm pending-changes apply` command. This will restart Tableau Server.

### Database Drivers

Tableau connectors require a driver to talk to the database. Before you can connect to data sources from Tableau Server, you must install drivers for the data sources you want to connect to. You can find information about supported data sources for Tableau Server on Linux on the [Tableau Server tech specs page](http://example.com). You can find driver links and installation instructions for all the supported connectors on the [Driver Download page](http://example.com).

**Important:** You must install the PostgreSQL driver if you want to use the built-in administrative views.
Install drivers in a cluster

You need to install the drivers for your data sources on the initial node in a Tableau Server cluster. If you install Tableau Server on multiple nodes, you must also install drivers on any node that runs any of the following processes:

- Application Server (Vizportal)
- Backgrounder
- Data Server
- VizQL Server

Server Crash Reporter

The Tableau Server administrator can enable an option to allow logs and related files to be sent to Tableau when the server has an issue that results in a crash. These files are used by Tableau to identify and address issues that cause crashes. By default this option is disabled, and it should only be enabled in organizations that are not subject to regulations related to data privacy.

**Important:** Do not enable crash reporting if your data is subject to privacy regulations.

If Tableau Server has a problem that results in a crash, log files and dump files are generated. If the crash data upload feature is enabled, these files are automatically gathered and zipped into an encrypted package that is sent in the background, at the scheduled time. The encrypted package is sent in small pieces to limit impact to network performance. Only one crash report is packaged and uploaded at a time (a new crash report is not packaged until the previous package has been uploaded) and is sent in a "first in, first out" order. You can schedule the sending for a low-use window to further reduce any impact to your users.

The encrypted package is made up crash dump files and logs that include the following:

- Crash/core dump files
- Error log files related to the crash
- Manifest files related to the crash

The files can contain data that includes:

- Machine-specific information (for example: hardware, operating system, domain).

- A snapshot of the contents of memory at the time of the crash, including application activity details like information about data connections, actions taken by the user in Tableau, and data being worked on in Tableau.

- Tableau information including customer-identifiable information.

Configure Server Crash Reporter

Server crash reporting is disabled by default. This topic describes how to enable and configure server crash reporting. Crash reports are encrypted and sent to Tableau. See Server Crash Reporter for more information.

Important: Do not enable crash reporting if your data is subject to privacy regulations.

Use the TSM web interface

1. Open TSM in a browser:

   https://<tsm-computer-name>:8850. For more information, see Sign in to Tableau Services Manager Web UI.

2. Click the Maintenance tab.

3. Under Other Maintenance Tasks, in Server Crash Reporter, select Enable crash reporting:
4. Specify the scheduled time of day to upload the crash reports to Tableau.

5. When you are finished, click **Pending Changes**, and then click **Apply Changes and Restart**.

**Use the TSM CLI**

Use the configuration file template below to create a json file. After you have filled in the options with the appropriate values, pass the json file and apply settings with the following commands:

```sh
tsm settings import -f path-to-file.json

tsm pending-changes apply
```

The `pending-changes apply` command displays a prompt to let you know this will restart Tableau Server if the server is running. The prompt displays even if the server is stopped, but in that case there is no restart. You can suppress the prompt using the `--ignore-prompt` option, but this does not change the restart behavior. For more information, see `tsm pending-changes apply`.

Because the configuration file is using `configKey` class, the values that you pass are not validated by TSM as they are when you use `configEntities` class. You can verify and set individual options by using the `tsm configuration` commands.
Crash reporter settings

The crash reporter settings in the template below specify a range of options for configuring Tableau Server to send crash reports to Tableau.

Configuration template

Use this template to configure the gateway settings.

For more explanation about configuration files, entities, and keys see Configuration File Example.

```json
{
   "configKeys": {
      "servercrashupload.enabled": "true",
      "servercrashupload.scheduled_time": "1:00:00 UTC",
      "servercrashupload.proxy_server_host": "",
      "servercrashupload.proxy_server_port": "",
      "servercrashupload.proxy_server_username": "",
      "servercrashupload.proxy_server_password": "",
      "servercrashupload.preserve_upload_packages": "false",
      "servercrashupload.delete_completed_dumps": "false"
   }
}
```

Configuration file reference

This table includes keys that you can set to configure crash reporting.

servercrashupload.enabled

**Default:** false.

Set to **true** to enable crash reporting.

servercrashupload.scheduled_time

**Default:** 1:00:00 UTC
Specifies the scheduled time that crash uploads will begin. Enter time of day in 24 hour format.

`servercrashupload.proxy_server_host`

If your organization uses a proxy server to communicate with the internet, specify the host name.

`servercrashupload.proxy_server_port`

If your organization uses a proxy server to communicate with the internet, specify the port number.

`servercrashupload.proxy_server_username`

If your proxy server requires authentication, specify the user name with this key.

`servercrashupload.proxy_server_password`

If your proxy server requires authentication, specify the password with this key.

`servercrashupload.preserve_upload_packages`

**Default:** false.

To save all packages that are created for a crash reporting, set this key to `true`.

By default, packages are saved to `/var/opt/tableau/tableau_server-/data/tabsvc/clustercontroller/tabcrashreporter`.

`servercrashupload.delete_completed_dumps`

**Default:** false.

To delete all dumps after they are sent, set this key to `true`. 
Navigate the Admin Areas of the Tableau Web Environment

As an administrator on Tableau Server or Tableau Online, you can access admin settings that aren’t available to other users to configure sites, users, projects, and to do other content-related tasks.

The settings in this article refer to the Tableau web environment. Tableau Server administrators with appropriate credentials can also change server settings such as processor, caching, authentication, distributed deployment, and related configurations using the TSM web environment. For information, see Sign in to Tableau Services Manager Web UI.

Access based on site role and number of sites

The menus you get when you sign in to Tableau Server or Tableau Online depend on the following conditions:

- Whether you’re a site or server administrator.

  Site administrator access is available on Tableau Online and Tableau Server. Server administrator access is only on Tableau Server.

- Whether you have access to only one site or to multiple sites.
On a **single-site** server, the site selector does not appear, and all other menus are the same.

In a **multi-site** environment, menus along the left enable you to modify a specific site or all sites, and to configure users, groups, schedules, tasks, and server settings.

To access server administrator settings that affect all sites, open the site menu by clicking the arrow next to the current site name, and then select **Manage all sites**.

The **Content** and **Group** tabs go away, and the site menu text changes to **All Sites** to let you know you are managing server-wide settings, and options like **Server Status** reflect the server-wide view.
To return to the site administration menus, select **All Sites**, and the select the site you want to manage.
Site administrator

If you are a site administrator for Tableau Online or Tableau Server, and you have access to multiple sites, you’ll get menus for selecting which site to manage, and for managing that site’s content, users, groups, schedules, and tasks, and for monitoring its status.

The site selector displays the name of the current site. To go to another site, select the site menu, and then select the site name.

If you have access to only one site, the site selector does not appear, but all other menus are the same.

Server administrator tasks

Server administrators (available with Tableau Server Enterprise only) can do the following:

- Monitor server status and activity.
- Generate log files.
- Add sites and edit site settings. Only server administrators can add sites to the server.
- Add users to the server, and assign users to sites.
- Add and manage site groups.

To manage settings only for a specific site, you must first navigate to the site. Within each site, you can do the following:
Site administrator tasks

A site administrator on Tableau Online or Tableau Server can do the following tasks:

- Administer content: Create projects, move content from one project to another, assign permissions, change ownership of a content resource, and so on.

- View, manage, and manually run schedules for extract refreshes and subscriptions.

- Add and manage site users (if allowed by the server administrator; see Add or Update Sites).

- Add and manage site groups.

- Monitor site activity.
Distributed and High Availability Tableau Server Installations

The most basic way to run Tableau Server is to install a single node. With this type of installation you have a fully functional Tableau Server, with all Tableau Services Manager (TSM) and Tableau Server processes running on that single node, but this may not be the optimal way to use Tableau Server. You can decide how to install Tableau based on your organization's needs, and your resources, adding additional nodes and configuring Tableau for high availability. Your installation options include:

- **Single-node installation**—This type of installation is reasonable for testing, running trials, and for environments that can handle occasional downtime and system availability due to lack of redundancy. All server processes are running on a single machine. There is less redundancy and fewer safeguards in the event of a problem with one of the server processes. You also need to make sure the computer you install Tableau Server on has adequate resources to handle the processes and the demands of users and data.

- **Distributed installation**—This type of installation is also called a multi-node installation and requires multiple computers so you can install and run server processes on those distributed nodes. Spreading the server processes out over multiple nodes can extend the reliability and efficiency of Tableau Server by providing redundancy and additional computing power. With the right configuration, a distributed installation can also provide you with automatic repository failover. For more information on failover, see Repository Failover.

- **Highly available (HA) installation**—An HA installation of Tableau Server is a special type of multi-node installation with a minimum of three nodes and multiple instances of key processes (the Repository, File Store/Data Engine (Hyper), and Coordination Service) on different computers. With an HA installation, there is built-in redundancy of those key processes, including multiple File Stores, and automatic
Repository failover. The goal is to minimize system downtime by eliminating single points of failure, and enabling detection of failures with failover where possible.

The first computer you install Tableau on, the "initial node," has some unique characteristics. Two processes run only on the initial node and cannot be moved to any other node except in a failure situation, the License service (License Manager) and TSM Controller (Administration Controller). Tableau Server includes a script that automates moving these two processes to one of your other existing nodes so you can get complete access back to TSM and keep Tableau Server running.

Two other processes are initially included on the initial node but can be added or moved to additional nodes, the CFS (Client File Service) and the Coordination Service. Depending on how your installation was configured with CFS and Coordination Service, you may also need to take steps to redeploy these.

For information about moving the License service and TSM Controller from the initial node to another node, see Recover from an initial node failure below.

Prerequisite

These instructions assume that your cluster meets the Distributed Requirements.

Note: All nodes in a multi-node cluster must have the same type of operating system. You cannot install a multi-node instance of Tableau Server on a combination of Linux and Windows nodes.

Licensing

You must have a valid Tableau Server product key. The type of license you have may determine how many nodes you can install Tableau on. You cannot use a trial license key for a distributed Tableau Server installation. For more information on licensing, see Licensing Overview.
Creating a distributed Tableau Server installation

These are the general steps you follow to create a distributed installation of Tableau Server:

1. Begin by installing Tableau Server on your initial node.
   For details, see Install and Configure Tableau Server.

2. Generate a node configuration (bootstrap) file on the initial node.
   For details, see Generate the node bootstrap file.

3. Install Tableau Server on an additional node using the node bootstrap file.
   For more information, see Install and initialize an additional node.

4. Configure your additional node with the processes you want to run on it.
   For more information, see Configure the additional node.

5. Repeat Steps 3 and 4 for any additional nodes you want to install.

6. Deploy a new Coordination Service ensemble.
   For more information, see Deploy a Coordination Service Ensemble.

Creating a highly available (HA) Tableau Server installation

A high availability Tableau Server installation is a special type of distributed installation, designed to accommodate failure in key server components without loss of complete server functionality. To create an HA installation, follow the same steps you take to create a distributed deployment but include additional steps to make the deployment highly available. These additional steps include adding at least two additional nodes (for a minimum of three nodes in the cluster), adding a second instance of the repository, and second instances of the data engine/file store, adding additional gateway processes, and deploying a Coordin-
You can also add a load balancer to distribute requests among the gateways.

At a high level, these are the steps you follow to create a highly available installation of Tableau Server:

1. Begin by installing Tableau Server on your initial node.
   
   For details, see Install and Configure Tableau Server.

2. Generate a node configuration (bootstrap) file on the initial node.
   
   For details, see Generate the node bootstrap file.

3. Install Tableau Server on at least two additional nodes using the node bootstrap file.
   
   For more information, see Install and initialize an additional node.

4. Configure each additional node with the processes you want to run on it. These must include a second copy of the Tableau Server repository, and a second copy of the data engine and file store, as well as additional instances of the gateway.
   
   For more information, see Configure the additional node.

5. Deploy a Coordination Service ensemble.
   
   For more information, see Deploy a Coordination Service Ensemble.

6. (Optional) Configure a load balancer.
   
   For more information, see Add a Load Balancer.

For details on how to create a three-node HA installation, see Example: Install and Configure a Three-Node HA Cluster.
Recover from an initial node failure

With a Tableau Server installation, the initial node includes two services that are only installed on that node, the License service, and the TSM Controller. If there is a problem with the initial node, Tableau Server may not continue to function, even when configured for high availability. To recover from a situation where the initial node fails, you can move the TSM Controller and the License service to one of your already configured nodes. This allows you to recover from the failure while using resources you already have in the cluster. You do not have to configure a standby initial node in case the initial node fails.

For details on how to recover from a failure on the initial node, see Recover from an Initial Node Failure.

Configure Coordination Service ensemble on additional nodes

Configuring a Coordination Service on multiple nodes provides additional duplication of processes and so reduces the possibility of server downtime due to an issue with one of the Coordination Service nodes. For details on how to deploy a Coordination Service ensemble on your cluster, see Deploy a Coordination Service Ensemble.

Tableau Server service license check

A number of processes are installed when you install Tableau Server. Some of these processes are dependent on the existence of a valid Tableau Server license while other installed processes are not. The subset of Tableau Server that require a valid Tableau Server license are considered "licensed processes."

When a licensed process starts or restarts, the process checks with the Tableau Server License Manager service on the initial node to verify there is a valid license. When the License Manager validates the license, the process is fully functional and able to respond to requests from other Tableau Server processes. Once a licensed process has received
confirmation from the License Manager, the process does not need to reconfirm the license for 72 hours, or until the process restarts. If the process is not able to verify that it is licensed (if the primary node is unavailable, for example) it cannot run, but it continues to check for a valid license until it confirms the license. To see when the last licensing check occurred, look at the log files in the /var/opt/tableau/tableau_server-/data/tabsvc/logs/licenseservice directory. For more information about licensed processes, see Licensed processes.

Distributed Requirements

Before you start to configure a Tableau Server cluster, make sure you meet the following requirements.

Hardware

While the computers you use in your cluster must meet the requirements described in Before you install..., they do not need to be identical.

Hardware Guidelines for High Availability

Here are some guidelines for the systems you use for failover and high availability:

- **Failover—three computers**: To configure a cluster that provides failover support for the file store and repository processes, you need at least three computers or VMs: one for the initial Tableau Server node and two for additional nodes.

- **Multiple gateways—three computers and a load balancer**: Adding multiple Gateway processes to your Tableau Server installation and using a load balancer to automatically distribute requests to those gateways enhances the reliability of Tableau further. To configure a cluster that provides failover support and multiple gateways, you need to add a load balancer to front your three-node cluster.

- **Failover & multiple gateway support—three computers and a load balancer**:
To configure a cluster that provides the above plus support for multiple gateways, you need at least three computers or VMs, and a load balancer to front the cluster.

- **High availability—three computers and a load balancer**: To configure for high availability, you need the resources described above.

- **Initial computers**: If you configure for high availability, the initial Tableau Server node may be running few or no Tableau Server processes. Therefore, the computer that serves as the initial node does not need as many cores as the ones running your additional nodes. You will, however, need adequate disk space for backups because the initial computer is used during the database backup and restore processes. In addition to the amount of space needed for the backup file, you need temporary disk space roughly 10 times the size of the backup file (so if your backup is 4 GB, you should have about 40 GB of temporary disk space available).

**Software**

- All nodes in a distributed installation must be running the same version of Tableau Server.

- All nodes must be running the same distribution of Linux.

**Installation location**

When you start Tableau Services Manager (TSM) you can specify a non-default location for the data directory. This location must be the same on all nodes in your distributed Tableau Server installation.

**Networking and Ports**

- **Ports**: As with any distributed system, the computers or VMs you use need to be able to communicate with one another. For information on how Tableau Services Manager handles port mapping, see Tableau Services Manager Ports.

- **Same subnet**: Do not install a distributed system across multiple subnets. Latency
between subnets can cause problems with Tableau server, so all nodes in the cluster should be installed on the same subnet.

**Note:** When installing a cluster on AWS, this requirement does not apply. For details, see "Install Tableau on Amazon Web Services" in the Tableau Server Help.

- **Static IP addresses:** Any computer running Tableau Server, whether it's a single server installation or part of a cluster, must have a static IP address.

- **Discoverable:** Each node in the cluster must be discoverable from other node computers using DNS or a local host file.

- **Time zone and time:** Each node in the cluster must be in the same timezone, with their system clocks synchronized. This may happen automatically. For example, if your nodes are all in the Active Directory domain, the domain controller usually handles this. If you are not sure your cluster meets this requirement, consult with your internal IT experts.

**Best Practices**

Here are some things to keep in mind before you start to install and configure:

- **IP addresses or computer names:** As mentioned above, each computer in the cluster must use a static IP address.

- **Backup:** It’s a best practice to create a backup prior to making significant system changes. See Back up Tableau Server data for steps.

**SSL**

If you are planning to configure SSL for a highly available Tableau Server cluster with multiple gateways and a load balancer (learn more), make sure that the SSL certificate you use
was issued for the load balancer’s host name. See Configure SSL for External HTTP Traffic to and from Tableau Server for other details.

Distributed Installation Recommendations

When you add nodes to a Tableau Server installation, you must decide how many processes to run on each computer. This page provides some general recommendations that are intended only as a starting point.

In addition to these general recommendations, you should also:

- Understand how your organization uses Tableau Server and tune your configuration for your use case—for example, whether you want to optimize for user response or for extract refreshes.

- Perform thorough performance testing to identify the best places to adjust process configuration.

For more information on tailoring a Tableau Server installation to your organization’s needs, see Performance Tuning Examples.

For more information on the requirements for a distributed installation and for information on configuring additional nodes, see Distributed and High Availability Tableau Server Installations.

Recommendations for all installations

Although the computers that make up a Tableau Server cluster do not need to have identical hardware, they must all meet the same minimum system requirements. The recommendations on this page assume that the computers where you install Tableau Server have eight cores or more.

The following recommendations apply to all server configurations:
- Run Backgrounder processes on a dedicated computer if you plan on refreshing extracts frequently. Backgrounder processes are generally the most CPU intensive and can slow down other processes on the same computer.

  **Note:** In versions earlier than 10.5, the backgrounder process spun up temporary instances of Data Engine when refreshing extracts. In 10.5, a single instance of Data Engine is installed with the backgrounder during setup. The backgrounder uses this running instance of Data Engine when refreshing extracts.

- If you plan to refresh extracts frequently or if you plan to refresh large extracts, increase the number of processes for Backgrounder processes.

- Run VizQL processes on a different computer than Backgrounder processes. Having them on the same machine means that extract refreshes can affect user views.

- The instance of Data Engine installed on the node where File Store is installed is used for querying data for view requests. Consider separating the File Store process from the backgrounder processes to help minimize the backgrounder tasks from affecting user views.

- Locating the repository (pgsql) and the File Store on the same node as the Administration Controller can reduce the length of time it takes to back up Tableau Server. The Administration Controller is usually on the initial node, unless you have had an initial node failure and had to move the controller to another node.

  When the repository is co-located with the controller it eliminates or greatly reduces the need to compress and transfer data between nodes during the backup process. This is also true with the File Store, especially if your organization uses extracts heavily.
**Note:** In a distributed installation with three or more nodes, you can have a maximum of two repository instances (active and passive). You can also run Tableau Server with one repository, but doing this means there is no failover available for the repository. For more information, see Tableau Server Repository.

**Install and Configure Additional Nodes**

After you install Tableau Server on one computer (or node), the server is functional and ready for use, but it has no redundancy. If there is a problem with a process or a problem with the computer itself, Tableau Server may be unavailable. In addition, all processes are running on a single computer, so there can be contention for resources on that computer.

You can extend your Tableau Server installation by adding Tableau to additional nodes, creating a distributed installation. This article describes the general steps for installing Tableau Server on additional nodes and assumes you have already installed Tableau on an initial node. For more information on installing Tableau on the initial node, see Install and Initialize TSM.

If you are installing Tableau Server on multiple nodes, you should install and configure one node at a time. This makes it easier to troubleshoot any issues you might run into.

**Important:** You should add and configure additional nodes when you can fully complete the process by applying pending changes. Adding a node without finishing by applying pending changes can result in users being unable to log into Tableau Server.

**Use the TSM web interface**

Generate the node bootstrap file

1. Open TSM in a browser:

   https://<tsm-computer-name>:8850
For more information, see Sign in to Tableau Services Manager Web UI.

2. Click the **Configuration** tab, and in the Add a Node box, click **Download Bootstrap File**.

![Tableau Server Topology](image)

The bootstrap file is created and copied to your local computer.

Beginning with version 2019.3, embedded credentials are included in the bootstrap file by default. If you don't want credentials embedded in the bootstrap file, clear the **Include temporary credentials in file** option. See Support for Embedded Credentials in Bootstrap Files in What's New for more information. If you want to completely disable the ability to include embedded credentials in node bootstrap files, you can set a configuration option for the server. See features.PasswordlessBootstrapInit for more details.
Install and initialize an additional node

Before you begin, verify that your node bootstrap file is recent. For example, if you have run
tsm security regenerate-internal-tokens after you generated a bootstrap file, then initialization will fail.

1. Copy the original installer you used on the first computer along with the bootstrap file you generated and put them in a location accessible from the new computer you are adding Tableau Server to. This could be a mounted network share, or directly on the new computer.

2. If you are running a local firewall, then you need to configure firewall rules for all the nodes in the cluster. For more information, see Configure Local Firewall.

3. On the new node, run the Tableau Server Setup program:

Use the package manager to install the Tableau Server package.

- On RHEL-like distributions, including CentOS, you have the option to install Tableau to a non-default location.

  - **Default location**—To install to the default location (/opt/tableau/tableau_server), run the following commands, where
    <version> is formatted as major-minor-maintenance (ex: 2019-2-5):

    ```
    sudo yum update
    sudo yum install tableau-server-<version>.x86_64.rpm
    ```

  - **Non-default location**—To install to a non-default location, you must use
    `rpm -i`. You will also need to install all dependent packages. See the note below.

    Run the following command:
sudo rpm -i --prefix /preferred/install/path

tableau-server.rpm

**Note:** When you use `yum` to install Tableau Server, all dependent packages are automatically downloaded and installed. This is the preferred method for installing Tableau. If you want to install to a non-default location, or your organization does not allow you to use `yum` and you must install using `rpm -i`, you must also install all dependent packages separately. For information about installing dependent packages, see Installing Tableau Server on an Air-Gapped Computer Running Linux.

- On Ubuntu and Debian, run the following commands, where `<version>` is formatted as major-minor-maintenance (ex: `2019-2-5`):

```
sudo apt-get update
sudo apt-get upgrade
sudo apt-get -y install gdebi-core
sudo gdebi -n tableau-server-<version>_<amd64>.deb
```

4. To initialize communications between your new node and the initial node, run the initialize-`tsm` script that is installed when you install Tableau Server.

Navigate to the `scripts` directory:

```
cd /opt/tableau/tableau_server/packages/scripts.<version_code>/
```

5. Run the initialize-`tsm` script:

```
sudo ./initialize-`tsm` -b /path/to/<bootstrap>.json --accepteula
```

- Use the `-b` flag to provide the path to the bootstrap file that you copied to the computer. If you have encrypted the bootstrap file, then you must pipe the file
as described in Securing secrets for import and export operations.

- If the bootstrap file was generated without embedded credentials, use the \texttt{-u} flag to specify the user name of the administrative user on the initial node. This is the name of an administrative user on the computer, not the Tableau Server administrator. You will be prompted for the user password. For more information, see tsm topology nodes get-bootstrap-file.

\begin{verbatim}
sudo ./initialize-tsm -b /path/to/<bootstrap>.json -u <admin-user-on-first-node> --accepteula
\end{verbatim}

\textbf{Important:} You must enter the credentials of the same user that you used during the installation process on the initial node.

- The \texttt{--accepteula} flag accepts the Tableau Server End User License Agreement (EULA). The EULA is available in the following location:

\begin{verbatim}
/opt/tableau/tableau_server/packages/docs.<version_code>/EULA.rtf
\end{verbatim}

Configure the additional node with Cluster Controller

1. Open TSM in a browser:

\begin{verbatim}
https://<tsm-computer-name>:8850
\end{verbatim}

For more information, see Sign in to Tableau Services Manager Web UI.

2. Click the \textbf{Configuration} tab. A message should tell you that the new node was added.

Tableau Server on Linux Administrator Guide
3. Cluster Controller is part of the minimum topology and is already selected. Each node must run Cluster Controller.

If you only want to set the minimum topology for the node, go to Step 4 below. You might do this if you will be adding additional nodes and are not ready to configure this node beyond the minimum.

If you want to add additional processes to the minimum topology, specify the processes that should run on the node. Do this by selecting the processes you want, or specifying the number of processes that should run on the node.

For example, to add a Gateway and two instances of the Backgrounder on the node:

   a. Select **Gateway**.

   b. Set the **Backgrounder** count to 2.

Adding Backgrounder to a node will also add an instance of Data Engine if one is not already on the node.

The specific processes and process counts you set will depend on your organizational environment and needs. Some processes are added automatically when
you add another process. For more information, see Configure Nodes and Tableau Server Processes.

**Note:** The TSM Web UI limits you to a maximum of 8 instances of processes that allow you to select the number of instances. To configure more instances than this, use the command line and the `TSM topology set-process` command. For more information, see `tsm topology set-process`.

4. Click **Pending Changes** at the top of the page:

If you are configuring a cluster with three or more nodes, a Coordination Service ensemble warning displays. You can continue. You will deploy a Coordination Service ensemble in a separate step.

5. Click **Apply Changes and Restart** and **Confirm** to confirm a restart of Tableau Server.

After Tableau Server restarts, the node is included with the minimum topology necessary and any additional processes you configured.

**Use the TSM CLI**

Generate the node bootstrap file

To install Tableau Server on additional computers you use the same installer you did for the initial node, along with a "bootstrap" file you generate on the initial node.

**Important:** The bootstrap file contains a copy of the master keystore file used for encrypting the configuration secrets. The file can also embedded credentials which are
valid for a predetermined amount of time (see `tabadmincontroller.auth.expiration.minutes`) and serve as a session cookie. We strongly recommend that you take additional measures to secure the bootstrap file using mechanisms as described in Securing secrets for import and export operations.

1. After installing Tableau Server on the initial node, generate the node bootstrap file.

2. On the initial node, open a terminal session.

3. Type this command to generate the bootstrap file:

   ```
   tsm topology nodes get-bootstrap-file --file <path\file>.json
   ```

   Beginning with version 2019.3, embedded credentials are included in the bootstrap file by default. If you don't want the bootstrap file to include credentials, use the `--no-embedded-credentials` option:

   ```
   tsm topology nodes get-bootstrap-file --file <path\file>.json --no-embedded-credentials.
   ```

   See Support for Embedded Credentials in Bootstrap Files in What's New for more information. If you want to completely disable the ability to include embedded credentials in node bootstrap files, you can set a configuration option for the server. See features.PasswordlessBootstrapInit for more details.

Install and initialize an additional node

1. Copy the original installer you used on the first computer along with the bootstrap file you generated and put them in a location accessible from the new computer you are adding Tableau Server to. This could be a mounted network share, or directly on the new computer.

2. If you are running a local firewall, then you need to configure firewall rules for all the
nodes in the cluster. For more information, see Configure Local Firewall.

3. On the new node, run the Tableau Server Setup program:

Use the package manager to install the Tableau Server package.

- On RHEL-like distributions, including CentOS, you have the option to install Tableau to a non-default location.

  - **Default location**—To install to the default location (/opt/tableau/tableau_server), run the following commands, where <version> is formatted as major-minor-maintenance (ex: 2019-2-5):

    ```
    sudo yum update
    sudo yum install tableau-server-<version>.x86_64.rpm
    ```

  - **Non-default location**—To install to a non-default location, you must use `rpm -i`. You will also need to install all dependent packages. See the note below.

    Run the following command:

    ```
    sudo rpm -i --prefix /preferred/install/path tableau-server.rpm
    ```

**Note:** When you use `yum` to install Tableau Server, all dependent packages are automatically downloaded and installed. This is the preferred method for installing Tableau. If you want to install to a non-default location, or your organization does not allow you to use `yum` and you must install using `rpm -i`, you must also install all dependent packages separately. For information about installing dependent packages, see Installing Tableau Server on an Air-Gapped Computer Running Linux.
- On Ubuntu and Debian, run the following commands, where `<version>` is formatted as major-minor-maintenance (ex: 2019-2-5):

  ```
sudo apt-get update
sudo apt-get upgrade
sudo apt-get -y install gdebi-core
sudo gdebi -n tableau-server-<version>_amd64.deb
```

4. To initialize communications between your new node and the initial node, run the initialize-tsm script that is installed when you install Tableau Server.

   On the new node:

   **Navigate to the `scripts` directory:**

   ```
cd /opt/tableau/tableau_server/packages/scripts.<version_code>/
```

5. Run the initialize-tsm script:

   ```
sudo ./initialize-tsm -b /path/to/<bootstrap>.json --accepteula
   ```

   - Use the `--b` flag to provide the path to the bootstrap file that you copied to the computer. If you have encrypted the bootstrap file, then you must pipe the file as described in Securing secrets for import and export operations.

   - If the bootstrap file was generated without embedded credentials, use the `--u` flag to specify the user name of the administrative user on the initial node. This is the name of an administrative user on the computer, not the Tableau Server administrator. You will be prompted for the user password. For more information, see tsm topology nodes get-bootstrap-file.

     ```
sudo ./initialize-tsm -b /path/to/<bootstrap>.json --u <admin-user-on-first-node> --accepteula
   ```
**Important:** You must enter the credentials of the same user that you used during the installation process on the initial node.

- The `--accepteula` flag accepts the Tableau Server End User License Agreement (EULA). The EULA is available in the following location:

  `/opt/tableau/tableau_server/packages/docs.<version_code>/EULA.rtf`

Configure the additional node

**Note:** This basic example illustrates how to set the topology on a node. For a more detailed, working multi-node example, see Example: Install and Configure a Three-Node HA Cluster.

On the initial (original) node, set the topology for the newly added node. The topology specifies which processes should run on the node, and how many instances of each process should run. The topology for the node will depend on your environment and organizational needs. The below are just examples of setting the topology.

1. On the new node, open a terminal session.

2. Get the node-id for the new node:

   `tsm topology list-nodes -v`

   The `-v` option lists the nodes and the processes they are currently running. You can identify the newly added node because it will not have any processes on it.

3. Specify the individual processes that should run on this node.

   Do this with the following command:
You must add an instance of the Cluster Controller to each node.

For example, to add the Cluster Controller, two instances of the Backgrounder, and a Gateway to node2:

```bash
tsm topology set-process -n node2 -pr clustercontroller -c 1
tsm topology set-process -n node2 -pr backgrounder -c 2
tsm topology set-process -n node2 -pr gateway -c 1
```

The specific processes and process counts you set will depend on your organizational environment and needs. Some processes are added automatically when you add another process. For more information, see Configure Nodes and Tableau Server Processes.

4. **Apply the node configuration.** The `pending-changes apply` command displays a prompt to let you know this will restart Tableau Server if the server is running. The prompt displays even if the server is stopped, but in that case there is no restart. You can suppress the prompt using the `--ignore-prompt` option, but this does not change the restart behavior. For more information, see `tsm pending-changes apply`.

```bash
tsm pending-changes apply
```

A warning about deploying a Coordination Service ensemble displays if you have deployed a three- or five-node cluster. If this is the only warning, you can safely override it using the `--ignore-warnings` option to apply the configuration changes in spite of the warning.

```bash
tsm pending-changes apply --ignore-warnings
```

**Install drivers**

You need to install drivers so that Tableau Server can connect to data and run extracts. Install these drivers on nodes that are running any of the following processes:
Tableau Server on Linux Administrator Guide

- VizQL Server (vizqlserver)
- Application Server (vizportal)
- Data Server (dataserver)
- Backgrounder (backgrounder)

Drivers and administrative views

If you want to use the built-in administrative views in Tableau Server, you also need to install the PostgreSQL driver on any nodes running any of the above processes.

For more information, see Database Drivers.

Database Drivers

Tableau connectors require a driver to talk to the database. Before you can connect to data sources from Tableau Server, you must install drivers for the data sources you want to connect to. You can find information about supported data sources for Tableau Server on Linux on the Tableau Server tech specs page. You can find driver links and installation instructions for all the supported connectors on the Driver Download page.

**Important:** You must install the PostgreSQL driver if you want to use the built-in administrative views.

Install drivers in a cluster

You need to install the drivers for your data sources on the initial node in a Tableau Server cluster. If you install Tableau Server on multiple nodes, you must also install drivers on any node that runs any of the following processes:

- Application Server (Vizportal)
- Backgrounder
- Data Server
- VizQL Server
Example: Install and Configure a Three-Node HA Cluster

A multi-node high availability installation of Tableau Server can help to maximize the efficiency and availability of Tableau Server. When configuring a multi-node Tableau Server deployment, the steps you perform are designed to build in redundancy, helping to reduce your potential downtime. In addition to simply improving efficiency by moving or adding server processes to additional nodes, you can create a highly available (HA) installation of Tableau Server by satisfying these requirements:

- Add additional nodes for a total of at least three nodes
- Deploy a Coordination Service ensemble on at least three nodes
- Add a second instance of the File Store on one of the additional nodes (Data Engine will be installed automatically, if it is not already on the node)
- Add a second instance of the Repository (pgsql) on one of the additional nodes

A Tableau Server installation that includes these additions will have built-in redundancy and can support failover in the event of a problem with the repository. This example shows how to do this, and more.

A Single Server System

After installing Tableau Server on an initial node, you have a system that is running everything it needs to function. It has at least one instance of all server processes and is the most basic configuration of Tableau Server. It has no redundancy. The server topology looks like this (some TSM-specific processes are not shown):
A Three-Node System

To build in redundancy, you need to add additional nodes to host instances of the repository and the data engine and file store processes. You can add instances of other processes, including multiple instances of a process on a node. In addition, to reduce the system’s vulnerability, you can run multiple gateways and additional instances of some of the server processes. The fewest number of computers required to achieve this configuration is three.

In the diagram below, the file store process has been added to both additional nodes. A second, passive instance of the repository has also been added to one of the other new nodes. Finally, the server processes (shown in blue) have been added to the additional nodes to provide redundancy.
Configuration steps

This procedure describes how to configure a three-node HA Tableau Server cluster with two repository instances and two file store/data engine instances on the additional nodes as
Before you begin

Before you install Tableau Server on any additional nodes, ensure that each additional node meets the distributed requirements. See Distributed Requirements for details.

Use the TSM web interface

**Note:** This operation includes steps that you may need to perform using the TSM command line.

Step 1: Install the initial node

See Install and Configure Tableau Server.

Step 2: Generate a bootstrap file for the additional nodes

1. Open TSM in a browser:

   https://<tsm-computer-name>:8850

   For more information, see Sign in to Tableau Services Manager Web UI.

2. Click the **Configuration** tab, and in the Add a Node box, click **Download Bootstrap File**.
The bootstrap file is created and copied to your local computer.

Beginning with version 2019.3, embedded credentials are included in the bootstrap file by default. If you don’t want credentials embedded in the bootstrap file, clear the Include temporary credentials in file option. See Support for Embedded Credentials in Bootstrap Files in What’s New for more information. If you want to completely disable the ability to include embedded credentials in node bootstrap files, you can set a configuration option for the server. See features.PasswordlessBootstrapInit for more details.

Step 3: Install and initialize an additional node

1. Copy the original installer you used on the first computer along with the bootstrap file you generated and put them in a location accessible from the new computer you are adding Tableau Server to. This could be a mounted network share, or directly on the
new computer.

2. If you are running a local firewall, then you need to configure firewall rules for all the nodes in the cluster. For more information, see Configure Local Firewall.

3. On the new node, run the Tableau Server Setup program:

Use the package manager to install the Tableau Server package.

- On RHEL-like distributions, including CentOS, you have the option to install Tableau to a non-default location.
  
  - **Default location**—To install to the default location (/opt/tableau/tableau_server), run the following commands, where `<version>` is formatted as major-minor-maintenance (ex: 2019-2-5):

    
    ```
    sudo yum update
    
sudo yum install tableau-server-<version>.x86_64.rpm
    ```

  - **Non-default location**—To install to a non-default location, you must use `rpm -i`. You will also need to install all dependent packages. See the note below.

    Run the following command:

    ```
    sudo rpm -i --prefix /preferred/install/path tableu-server.rpm
    ```

    **Note:** When you use `yum` to install Tableau Server, all dependent packages are automatically downloaded and installed. This is the preferred method for installing Tableau. If you want to install to a non-default
location, or your organization does not allow you to use `yum` and you must install using `rpm -i`, you must also install all dependent packages separately. For information about installing dependent packages, see Installing Tableau Server on an Air-Gapped Computer Running Linux.

- **On Ubuntu and Debian**, run the following commands, where `<version>` is formatted as major-minor-maintenance (ex: 2019-2-5):

  ```
  sudo apt-get update
  sudo apt-get upgrade
  sudo apt-get -y install gdebi-core
  sudo gdebi -n tableau-server-<version>_amd64.deb
  ```

4. To initialize communications between your new node and the initial node, run the initialize-tsm script that is installed when you install Tableau Server.

   Navigate to the scripts directory:

   ```
   cd /opt/tableau/tableau_server/packages/scripts.<version_code>/
   ```

5. Run the initialize-tsm script:

   ```
   sudo ./initialize-tsm -b /path/to/<bootstrap>.json -- accepteula
   ```

   - Use the `-b` flag to provide the path to the bootstrap file that you copied to the computer. If you have encrypted the bootstrap file, then you must pipe the file as described in Securing secrets for import and export operations.

   - If the bootstrap file was generated without embedded credentials, use the `-u` flag to specify the user name of the administrative user on the initial node. This is the name of an administrative user on the computer, not the Tableau Server.
You will be prompted for the user password. For more information, see tsm topology nodes get-bootstrap-file.

```
sudo ./initialize-tsm -b /path/to/<bootstrap>.json -u <admin-user-on-first-node> --accepteula
```

**Important:** You must enter the credentials of the same user that you used during the installation process on the initial node.

- The `--accepteula` flag accepts the Tableau Server End User License Agreement (EULA). The EULA is available in the following location:

```
/opt/tableau/tableau_server/packages/docs.<version_code>/EULA.rtf
```

### Step 4: Install and initialize a second additional node

1. Copy the original installer you used on the first computer along with the bootstrap file you generated and put them in a location accessible from the new computer you are adding Tableau Server to. This could be a mounted network share, or directly on the new computer.

2. If you are running a local firewall, then you need to configure firewall rules for all the nodes in the cluster. For more information, see Configure Local Firewall.

3. On the new node, run the Tableau Server Setup program:

   Use the package manager to install the Tableau Server package.

   - On RHEL-like distributions, including CentOS, you have the option to install Tableau to a non-default location.

     - **Default location**—To install to the default location (`/opt/tableau/tableau_server`), run the following commands, where `<version>` is formatted as major-minor-maintenance (ex:
sudo yum update

sudo yum install tableau-server-<version>.x86_64.rpm

**Non-default location**—To install to a non-default location, you must use `rpm -i`. You will also need to install all dependent packages. See the note below.

Run the following command:

```
sudo rpm -i --prefix /preferred/install/path
tableau-server.rpm
```

**Note:** When you use `yum` to install Tableau Server, all dependent packages are automatically downloaded and installed. This is the preferred method for installing Tableau. If you want to install to a non-default location, or your organization does not allow you to use `yum` and you must install using `rpm -i`, you must also install all dependent packages separately. For information about installing dependent packages, see Installing Tableau Server on an Air-Gapped Computer Running Linux.

- On Ubuntu and Debian, run the following commands, where `<version>` is formatted as major-minor-maintenance (ex: 2019-2-5):

  ```
sudo apt-get update
sudo apt-get upgrade
sudo apt-get -y install gdebi-core
sudo gdebi -n tableau-server-<version>_amd64.deb
  ```

4. To initialize communications between your new node and the initial node, run the
initialize-tsm script that is installed when you install Tableau Server.

Navigate to the scripts directory:

cd /opt/tableau/tableau_server/packages/scripts.<version_code>/

5. Run the initialize-tsm script:

    sudo ./initialize-tsm -b /path/to/<bootstrap>.json --accepteula

   • Use the -b flag to provide the path to the bootstrap file that you copied to the computer. If you have encrypted the bootstrap file, then you must pipe the file as described in Securing secrets for import and export operations.

   • If the bootstrap file was generated without embedded credentials, use the -u flag to specify the user name of the administrative user on the initial node. This is the name of an administrative user on the computer, not the Tableau Server administrator. You will be prompted for the user password. For more information, see tsm topology nodes get-bootstrap-file.

      sudo ./initialize-tsm -b /path/to/<bootstrap>.json -u <admin-user-on-first-node> --accepteula

      **Important:** You must enter the credentials of the same user that you used during the installation process on the initial node.

   • The --accepteula flag accepts the Tableau Server End User License Agreement (EULA). The EULA is available in the following location:

      /opt/tableau/tableau_server/packages/docs.<version_code>/EULA.rtf
Tableau Server on Linux Administrator Guide

Step 5: Configure the new node with a minimum topology

To complete the process of adding new nodes to your cluster, you need to configure them with a minimum topology. With a minimum topology, the only pending change will be the addition of Cluster Controller, which is required on each node. If you want other processes on the nodes you can add most of them at the same time. This example only configures the nodes with Cluster Controller.

1. Open TSM in a browser:

   https://<tsm-computer-name>:8850

   For more information, see Sign in to Tableau Services Manager Web UI.

2. Click the **Configuration** tab. A message should tell you that new nodes were added.

   ![Configuration tab screenshot]

   Click **Continue** to dismiss the message.

3. Click **Pending Changes** at the top of the page:

   ![Pending Changes screenshot]
If you are configuring a cluster with three or more nodes, a Coordination Service ensemble warning displays. You can continue. You will deploy a Coordination Service ensemble in a separate step.

4. Click **Apply Changes and Restart** and **Confirm** to confirm a restart of Tableau Server.

When Tableau Server restarts, the nodes are included with the minimum topology necessary.

**Step 6: Deploy a Coordination Service ensemble**

If you install a total of three or more nodes, you should also deploy a Coordination Service ensemble. If you do not, you will get a warning message every time you make changes to the server configuration or topology. You can ignore this message, but as a best practice you should deploy a multi-node Coordination Service ensemble.

When you install Tableau Server, a single instance of the Coordination Service is installed on the initial node. TSM and Tableau Server depend on the Coordination Service to function properly, so to provide redundancy and ensure availability on multi-node installations, configure additional instances of the Coordination Service by deploying a Coordination Service ensemble. Coordination Service ensembles are installed with one, three, or five instances of the Coordination Service. In a three-node installation of Tableau Server, the recommended number of Coordination Service instances is three, one on each node.

Do not attempt to deploy a Coordination Service ensemble if there are other changes pending. Discard or apply any pending changes before deploying a new Coordination Service ensemble.

**Note:** This operation includes steps that you may need to perform using the TSM command line.

The following steps illustrate how to deploy a new Coordination Service ensemble on an existing three-node Tableau Server cluster and clean up the old ensemble.
1. On the initial node, open a terminal session.

2. Type this command to sign in to Tableau Server as a TSM administrator:

   tsm login -u <username>

   You will be prompted for your password.

3. Type this command to stop Tableau Server:

   tsm stop

   Some TSM processes will continue to run, including the Administration Controller and Administration Agent.

4. Confirm there are no pending changes:

   tsm pending-changes list

   If there are pending changes, you need to either discard the changes or apply them.

   Applying pending changes will take some time:

   • Discard the changes
     
     tsm pending-changes discard

   or

   • Apply the changes:
     
     tsm pending-changes apply

   Wait until the command completes and you are returned to the system prompt.

5. Get the node IDs for each node in the cluster:

   tsm topology list-nodes -v

6. Use the tsm topology deploy-coordination-service command to add a
new Coordination Service ensemble by adding the Coordination Service to specified nodes. You must specify the node(s) that the Coordination Service should be added to. The command also makes the new ensemble the "production" ensemble (the ensemble in use).

For example, deploy the Coordination Service to all three nodes of a three-node cluster, where the nodes are node1, node2, and node3:

```bash
tsm topology deploy-coordination-service -n node1,node2,node3
```

7. Wait until the new Coordination Service ensemble is running and the server is ready for the next step.

**Important.** If you attempt to clean up the old Coordination Service ensemble before the server is in the proper state, you can put the server into an unrecoverable state and may need to completely reinstall Tableau.

a. Check the status of the server:

```bash
tsm status -v
```

If the deployment is not complete, you may see processes showing as running when they are not, and the Coordination Service showing a status of "unavailable" while the service is synchronizing between nodes on the cluster. Tableau Server may show as being in an error state while this is happening. You may also get an error message: "Could not connect to TSM Controller at '<host>:8850'." This is normal when the server is returning to a valid state.

b. Check the status of the server periodically until you are prompted to sign in again.

c. When you are prompted, sign in to TSM and continue to check the server
status until you see a status of "STOPPED" for each node. If the status of a node shows as "ERROR" you need to wait. When each node status is "STOPPED" you should also see the following services running:

On the initial node:

- Two instances of the Coordination Service on the initial node, both with a status of "running".
- The Administration Controller with a status of "running". (The Administration Controller is only installed on the initial node.)
- The Administration Agent with a status of "running".
- Additional services on the initial node, all with a status of "running": Service Manager, License Manager, Client File Service.

On the additional nodes:

- One or more instance of the Coordination Service on each additional node you specified when you deployed the new ensemble, all with a status of "running". If you are deploying a new ensemble to nodes that already had Coordination Service running, you will see two instances of the service.
- The Administration Agent on every node, with a status of "running".

If you do not see a status of "running" for all of the above, wait a few minutes and run the status command again.

**Note:** If there is a problem with an instance of the Coordination Service (if it shows as stopped for example), you can toggle back to your previous Coordination Service ensemble using the `tsm topology toggle-coordination-service` command. To do this, the rest of the services
should be in the state described above, including Administration Controller and Agent. You can toggle back to the previous ensemble only if you have not run the `cleanup-coordination-service` command. Tableau Server cannot be running when you use this command.

8. Remove the old ensemble. This step is required. You cannot run Tableau Server with multiple Coordination Service ensembles configured.

   `tsm topology cleanup-coordination-service`

   Tableau Server must be stopped when you use this command, but some TSM services will be running (see above).

9. Start Tableau Server:

   `tsm start`

   For more information and details on deploying a new Coordination Service ensemble, see Deploy a Coordination Service Ensemble.

Step 7: Configure processes for the second node

1. Open TSM in a browser:

   `https://<tsm-computer-name>:8850`

   For more information, see Sign in to Tableau Services Manager Web UI.

2. Click the **Configuration** tab.

3. Specify the processes and number of instances that should run on the first additional node.

   In this example:
Tableau Server on Linux Administrator Guide

a. Select **Gateway**.

b. Set the **Application Server** (vizportal) count to 2.

c. Set the **VizQL Server** count to 2.

d. Set the **Cache Server** count to 2.

e. Select **Search & Browse**.

f. Set the **Backgrounder** count to 2.

   Adding Backgrounder to a node will also add an instance of Data Engine if one is not already on the node.

g. Set the **Data Server** count to 2.

h. Select **File Store**.

i. Select **Repository** (pgsql).

The specific processes and process counts you set will depend on your organizational environment and needs. Some processes are added automatically when you add another process. For more information, see Configure Nodes and Tableau Server Processes.

**Note:** The TSM Web UI limits you to a maximum of 8 instances of processes that allow you to select the number of instances. To configure more instances than this, use the command line and the **TSM topology set-process command**. For more information, see tsm topology set-process.

Step 8: Configure processes for third node

1. In TSM, on the **Configuration** tab, specify the processes and number of instances that should run on the second additional node.

   In this example:
a. Select **Gateway**.

b. Set the **Application Server** (vizportal) count to 2.

c. Set the **VizQL Server** count to 2.

d. Set the **Cache Server** count to 2.

e. Select **Search & Browse**.

f. Set the **Backgrounder** count to 2.

Adding Backgrounder to a node will also add an instance of Data Engine if one is not already on the node.

g. Set the **Data Server** count to 2.

h. Select **File Store**.

2. Click **Pending Changes** at the top of the page:

   ![Pending Changes](image)

   The Pending Changes list displays.

3. Click **Apply Changes and Restart** and **Confirm** to confirm a restart of Tableau Server.

**Use the TSM CLI**

Step 1: Install the initial node

See **Install and Configure Tableau Server**.

Step 2: Generate a bootstrap file for the additional nodes

To install Tableau Server on additional computers you use the same installer you did for the initial node, along with a "bootstrap" file you generate on the initial node.
Important: The bootstrap file contains a copy of the master keystore file used for encrypting the configuration secrets. The file can also embed credentials which are valid for a predetermined amount of time (see `tabadmincontroller.auth.expiration.minutes`) and serve as a session cookie. We strongly recommend that you take additional measures to secure the bootstrap file using mechanisms as described in Securing secrets for import and export operations.

1. After installing Tableau Server on the initial node, generate the node bootstrap file.

2. On the initial node, open a terminal session.

3. Type this command to generate the bootstrap file:

   ```bash
tsm topology nodes get-bootstrap-file --file <pathile>.json
   ```

   Beginning with version 2019.3, embedded credentials are included in the bootstrap file by default. If you don’t want the bootstrap file to include credentials, use the `-nocred` or `--no-embedded-credentials` option:

   ```bash
tsm topology nodes get-bootstrap-file --file <pathile>.json --no-embedded-credentials.
   ```

   See Support for Embedded Credentials in Bootstrap Files in What’s New for more information. If you want to completely disable the ability to include embedded credentials in node bootstrap files, you can set a configuration option for the server. See features.PasswordlessBootstrapInit for more details.

Step 3: Install an additional node

1. Copy the original installer you used on the first computer along with the bootstrap file you generated and put them in a location accessible from the new computer you are adding Tableau Server to. This could be a mounted network share, or directly on the
new computer.

2. If you are running a local firewall, then you need to configure firewall rules for all the nodes in the cluster. For more information, see Configure Local Firewall.

3. On the new node, run the Tableau Server Setup program:

Use the package manager to install the Tableau Server package.

- On RHEL-like distributions, including CentOS, you have the option to install Tableau to a non-default location.

  - **Default location**—To install to the default location (/opt/tableau/tableau_server), run the following commands, where `<version>` is formatted as major-minor-maintenance (ex: 2019-2-5):

    ```
    sudo yum update
    sudo yum install tableau-server-<version>.x86_64.rpm
    ```

  - **Non-default location**—To install to a non-default location, you must use `rpm -i`. You will also need to install all dependent packages. See the note below.

    Run the following command:

    ```
    sudo rpm -i --prefix /preferred/install/path tableau-server.rpm
    ```

**Note:** When you use `yum` to install Tableau Server, all dependent packages are automatically downloaded and installed. This is the preferred method for installing Tableau. If you want to install to a non-default location, use `rpm -i` with the appropriate options.
location, or your organization does not allow you to use yum and you must install using rpm -i, you must also install all dependent packages separately. For information about installing dependent packages, see Installing Tableau Server on an Air-Gapped Computer Running Linux.

- On Ubuntu and Debian, run the following commands, where <version> is formatted as major-minor-maintenance (ex: 2019-2-5):

  ```
sudo apt-get update
sudo apt-get upgrade
sudo apt-get -y install gdebi-core
sudo gdebi -n tableau-server-<version>_amd64.deb
  ```

4. To initialize communications between your new node and the initial node, run the initialize-tsm script that is installed when you install Tableau Server.

   On the new node:

   Navigate to the scripts directory:

   ```
   cd /opt/tableau/tableau_server/packages/scripts.<version_code>/
   ```

5. Run the initialize-tsm script:

   ```
sudo ./initialize-tsm -b /path/to/<bootstrap>.json --accepteula
   ```

   - Use the -b flag to provide the path to the bootstrap file that you copied to the computer. If you have encrypted the bootstrap file, then you must pipe the file as described in Securing secrets for import and export operations.

   - If the bootstrap file was generated without embedded credentials, use the -u flag to specify the user name of the administrative user on the initial node. This
is the name of an administrative user on the computer, not the Tableau Server administrator. You will be prompted for the user password. For more information, see tsm topology nodes get-bootstrap-file.

```
sudo ./initialize-tsm -b /path/to/<bootstrap>.json -u <admin-user-on-first-node> --acceptheula
```

**Important:** You must enter the credentials of the same user that you used during the installation process on the initial node.

- The `--acceptheula` flag accepts the Tableau Server End User License Agreement (EULA). The EULA is available in the following location:

```
/opt/tableau/tableau_server/packages/docs.<version_code>/EULA.rtf
```

**Step 4: Install a second additional node**

Install Tableau Server on the second additional node:

1. Copy the original installer you used on the first computer along with the bootstrap file you generated and put them in a location accessible from the new computer you are adding Tableau Server to. This could be a mounted network share, or directly on the new computer.

2. If you are running a local firewall, then you need to configure firewall rules for all the nodes in the cluster. For more information, see Configure Local Firewall.

3. On the new node, run the Tableau Server Setup program:

   Use the package manager to install the Tableau Server package.

   - On RHEL-like distributions, including CentOS, you have the option to install Tableau to a non-default location.
Tableau Server on Linux Administrator Guide

- **Default location**—To install to the default location (/opt/tableau/tableau_server), run the following commands, where <version> is formatted as major-minor-maintenance (ex: 2019-2-5):

```
sudo yum update

sudo yum install tableau-server-<version>.x86_64.rpm
```

- **Non-default location**—To install to a non-default location, you must use `rpm -i`. You will also need to install all dependent packages. See the note below.

Run the following command:

```
sudo rpm -i --prefix /preferred/install/path tableau-server.rpm
```

**Note:** When you use `yum` to install Tableau Server, all dependent packages are automatically downloaded and installed. This is the preferred method for installing Tableau. If you want to install to a non-default location, or your organization does not allow you to use `yum` and you must install using `rpm -i`, you must also install all dependent packages separately. For information about installing dependent packages, see Installing Tableau Server on an Air-Gapped Computer Running Linux.

- On Ubuntu and Debian, run the following commands, where <version> is formatted as major-minor-maintenance (ex: 2019-2-5):

```
sudo apt-get update

sudo apt-get upgrade

sudo apt-get -y install gdebi-core

sudo gdebi -n tableau-server-<version>_amd64.deb
```
4. To initialize communications between your new node and the initial node, run the initialize-tsm script that is installed when you install Tableau Server.

On the new node:

**Navigate to the scripts directory:**

```bash
cd /opt/tableau/tableau_server/packages/scripts.<version_code>/
```

5. Run the initialize-tsm script:

```bash
sudo ./initialize-tsm -b /path/to/<bootstrap>.json --accepteula
```

- Use the `-b` flag to provide the path to the bootstrap file that you copied to the computer. If you have encrypted the bootstrap file, then you must pipe the file as described in Securing secrets for import and export operations.

- If the bootstrap file was generated without embedded credentials, use the `-u` flag to specify the user name of the administrative user on the initial node. This is the name of an administrative user on the computer, not the Tableau Server administrator. You will be prompted for the user password. For more information, see tsm topology nodes get-bootstrap-file.

```bash
sudo ./initialize-tsm -b /path/to/<bootstrap>.json -u <admin-user-on-first-node> --accepteula
```

**Important:** You must enter the credentials of the same user that you used during the installation process on the initial node.

- The `--accepteula` flag accepts the Tableau Server End User License Agreement (EULA). The EULA is available in the following location:
Step 5: Add a process to the additional nodes

1. On the initial node, configure a cluster controller instance on each additional node:

   tsm topology set-process -n <nodeID_second-node> -pr clustercontroller -c 1

   tsm topology set-process -n <nodeID_third-node> -pr clustercontroller -c 1

2. Apply the node configuration changes:

   tsm pending-changes apply --ignore-warnings

   A warning about deploying a Coordination Service ensemble displays because you have deployed a three-node cluster. Use the --ignore-warnings option to apply the configuration changes in spite of the warning. You will deploy a new Coordination Service ensemble in the next step.

Step 6: Deploy a Coordination Service ensemble

If you install a total of three or more nodes, you should also deploy a Coordination Service ensemble. If you do not, you will get a warning message every time you make changes to the server configuration or topology. You can ignore this message, but as a best practice you should deploy a multi-node Coordination Service ensemble.

When you install Tableau Server, a single instance of the Coordination Service is installed on the initial node. TSM and Tableau Server depend on the Coordination Service to function properly, so to provide redundancy and ensure availability on multi-node installations, configure additional instances of the Coordination Service by deploying a Coordination Service ensemble. Coordination Service ensembles are installed with one, three, or five instances of the Coordination Service. In a three-node installation of Tableau Server, the recommended number of Coordination Service instances is three, one on each node.
Do not attempt to deploy a Coordination Service ensemble if there are other changes pending. Discard or apply any pending changes before deploying a new Coordination Service ensemble.

**Note:** This operation includes steps that you may need to perform using the TSM command line.

The following steps illustrate how to deploy a new Coordination Service ensemble on an existing three-node Tableau Server cluster and clean up the old ensemble.

1. On the initial node, open a terminal session.

2. Type this command to sign in to Tableau Server as a TSM administrator:

   ```
   tsm login -u <username>
   ```

   You will be prompted for your password.

3. Type this command to stop Tableau Server:

   ```
   tsm stop
   ```

   Some TSM processes will continue to run, including the Administration Controller and Administration Agent.

4. Confirm there are no pending changes:

   ```
   tsm pending-changes list
   ```

   If there are pending changes, you need to either discard the changes or apply them. Applying pending changes will take some time:

   - Discard the changes
     ```
     tsm pending-changes discard
     ```
or

- Apply the changes:

  
  tsm pending-changes apply

  Wait until the command completes and you are returned to the system prompt.

5. Get the node IDs for each node in the cluster:

  
  tsm topology list-nodes -v

6. Use the `tsm topology deploy-coordination-service` command to add a new Coordination Service ensemble by adding the Coordination Service to specified nodes. You must specify the node(s) that the Coordination Service should be added to. The command also makes the new ensemble the "production" ensemble (the ensemble in use).

   For example, deploy the Coordination Service to all three nodes of a three-node cluster, where the nodes are node1, node2, and node3:

   
   tsm topology deploy-coordination-service -n node1,node2,node3

7. Wait until the new Coordination Service ensemble is running and the server is ready for the next step.

   **Important.** If you attempt to clean up the old Coordination Service ensemble before the server is in the proper state, you can put the server into an unrecoverable state and may need to completely reinstall Tableau.

   a. Check the status of the server:

   
   tsm status -v
If the deployment is not complete, you may see processes showing as running when they are not, and the Coordination Service showing a status of "unavailable" while the service is synchronizing between nodes on the cluster. Tableau Server may show as being in an error state while this is happening. You may also get an error message: "Could not connect to TSM Controller at '<host>:8850'." This is normal when the server is returning to a valid state.

b. Check the status of the server periodically until you are prompted to sign in again.

c. When you are prompted, sign in to TSM and continue to check the server status until you see a status of "STOPPED" for each node. If the status of a node shows as "ERROR" you need to wait. When each node status is "STOPPED" you should also see the following services running:

On the initial node:

- Two instances of the Coordination Service on the initial node, both with a status of "running".
- The Administration Controller with a status of "running". (The Administration Controller is only installed on the initial node.)
- The Administration Agent with a status of "running".
- Additional services on the initial node, all with a status of "running": Service Manager, License Manager, Client File Service.

On the additional nodes:

- One or more instance of the Coordination Service on each additional node you specified when you deployed the new ensemble, all with a status of "running". If you are deploying a new ensemble to nodes that already had Coordination Service running, you will see two instances of
the service.

- The Administration Agent on every node, with a status of "running".

If you do not see a status of "running" for all of the above, wait a few minutes and run the status command again.

**Note:** If there is a problem with an instance of the Coordination Service (if it shows as *stopped* for example), you can toggle back to your previous Coordination Service ensemble using the `tsm topology toggle-coordination-service` command. To do this, the rest of the services should be in the state described above, including Administration Controller and Agent. You can toggle back to the previous ensemble only if you have not run the `cleanup-coordination-service` command. Tableau Server cannot be running when you use this command.

8. Remove the old ensemble. This step is *required*. You cannot run Tableau Server with multiple Coordination Service ensembles configured.

   `tsm topology cleanup-coordination-service`

   Tableau Server must be stopped when you use this command, but some TSM services will be running (see above).

9. Start Tableau Server:

   `tsm start`

   For more information and details on deploying a new Coordination Service ensemble, see Deploy a Coordination Service Ensemble.
Step 7: Configure processes for the second node

1. On the initial node, set the topology for the second node. The topology specifies which processes should run on the node, and how many instances of each process should run. For details about configuring nodes, see Configure Nodes. Data engine will be added automatically when you add other processes. For details on when data engine is added, see Tableau Server Processes.
   
a. Get the node-id for the new node: `tsm topology list-nodes -v`
   
b. Specify individual processes that will run on the new node:

   ```
   tsm topology set-process -n <node-id> -pr gateway -c 1
   tsm topology set-process -n <node-id> -pr vizqlserver -c 2
   tsm topology set-process -n <node-id> -pr vizportal -c 2
   tsm topology set-process -n <node-id> -pr backgrounder -c 2
   tsm topology set-process -n <node-id> -pr cacheserver -c 2
   tsm topology set-process -n <node-id> -pr searchserver -c 1
   tsm topology set-process -n <node-id> -pr dataserver -c 2
   tsm topology set-process -n <node-id> -pr filestore -c 1
   tsm topology set-process -n <node-id> -pr postgres -c 1
   ```

2. Apply the node configuration changes. You will be prompted with a message that
Tableau Server on Linux Administrator Guide

Tableau Server will restart.

    tsm pending-changes apply

Step 8: Configure processes for third node

On the initial node, set the topology for the new node. The topology specifies which processes should run on the node, and how many instances of each process should run. Data engine will be added automatically when you add other processes. For details on when data engine is added, see Tableau Server Processes.

1. Get the node-id for the new node:

      tsm topology list-nodes -v

2. Specify individual processes that will run on the new node:

      tsm topology set-process -n <node-id> -pr gateway -c 1
      tsm topology set-process -n <node-id> -pr vizqlserver -c 2
      tsm topology set-process -n <node-id> -pr vizportal -c 2
      tsm topology set-process -n <node-id> -pr backgrounder -c 2
      tsm topology set-process -n <node-id> -pr cacheserver -c 2
      tsm topology set-process -n <node-id> -pr searchserver -c 1
      tsm topology set-process -n <node-id> -pr dataserver -c 2
      tsm topology set-process -n <node-id> -pr filestore -c 1

3. Apply the node configuration. You will be prompted with a message that Tableau Server will restart.

      tsm pending-changes apply
4. Start the server:

    tsm start

Step 9: Configure firewall rules (optional)

If you are running a local firewall, then you need to configure firewall rules for all the nodes in the cluster. For more information, see Configure Local Firewall. For more information, see Local firewall configuration.

Add a Load Balancer

At this point, all three nodes have gateways, which are used to route requests to available server processes. All gateways are active, but to further reduce the potential for downtime in the cluster, you can configure a load balancer. For more information, see Add a Load Balancer.

Add a Load Balancer

You can enhance the reliability of Tableau Server by running gateways on multiple nodes, and configuring a load balancer to distribute requests across the gateways. Unlike the repository process, which can be active or passive, all gateway processes are active. If one gateway in a cluster becomes unavailable, the load balancer stops sending requests to it. The load balancer algorithm you choose determines how the gateways will route client requests.

**Notes:** If you will be using Kerberos authentication, you need to configure Tableau Server for your load balancer before you configure Tableau Server for Kerberos. For more information, see Configure Kerberos.

Windows SSPI authentication is not supported with a load balancer.

Note the following as you configure your load balancer to work with Tableau Server:
Tested load balancers: Tableau Server clusters with multiple gateways have been tested with Apache and F5 load balancers.

If you are using an Apache load balancer and creating custom administrative views, you need to connect directly to the Tableau Server repository. You cannot connect through the load balancer.

Tableau Server URL: When a load balancer is in front of a Tableau Server cluster, the URL that’s accessed by Tableau Server users belongs to the load balancer, not the initial Tableau Server node.

Trusted host settings: The computer running the load balancer must be identified to Tableau Server as a trusted host. See the procedure below for how to configure Tableau Server.

Proxy server configurations: The settings used to identify a load balancer to Tableau Server are the same ones that are used to identify a proxy server. If your Tableau Server cluster requires both a proxy server and a load balancer, both must use a single external URL defined in `gateway.public.host` and all proxy servers and load balancers must be specified in `gateway.trusted` and `gateway.trusted_hosts`. For more information, see Configuring Proxies for Tableau Server.

Configure Tableau Server to work with a load balancer

Use the following key-value pairs to configure Tableau Server for a load balancer. You can set the values individually with the `tsm configuration set` command.

Or, use the configuration file template example at the end of this topic to create a `configKey json` file. After you have filled in the options with the appropriate values, pass the json file and apply settings with `tsm commands`.

Load balancer configuration key-value pairs

The key-value pairs described in this section can be set with the `tsm configuration set command`. After you have set all values, run the following command:
tsm pending-changes apply

gateway.public.host

**Value:** name

The value, name is the URL that will be used to reach Tableau Server through the load balancer.

For example, if Tableau Server is reached by entering `tableau.example.com` in a browser address bar, enter this command:

```bash
cvm configuration set -k gateway.public.host -v "tableau.example.com"
```

gateway.public.port

**Value:** port_number

By default, Tableau assumes that the load balancer is listening on port 80 for external communications. Use this key to specify a different port.

For example, if your load balancer is configured for SSL and listening on port 443, enter the following command:

```bash
cvm configuration set -k gateway.public.port -v "443"
```

gateway.trusted

**Value:** server

Specify the load balancer(s) IPv4 address or computer name.

The value for server can be a comma-separated list, for example:

```bash
cvm configuration set -k gateway.trusted -v "10.32.139.45, 10.32.139.46, 10.32.139.47"
```

or
tsm configuration set -k gateway.trusted -v "proxy1, proxy2, proxy3"

gateway.trusted_hosts

Value:

Specify alternate names for the load balancer(s), such as its fully-qualified domain name, any non-fully-qualified domain names, and any aliases. These are the names a user might type in a browser. Separate each name with a comma:

tsm configuration set -k gateway.trusted_hosts -v "name1, name2, name3"

For example:

tsm configuration set -k gateway.trusted_hosts -v "lb.example.com, lb, ftp.example.com, www.example.com"

Configuration template example

You can configure all of the settings in one operation by customizing the following configuration template example and creating a configKey json file.

Refer to the key-value pair descriptions above to customize the values for your organization.

```json
{
  "configKeys": {
    "gateway.public.host": "tableau.example.com",
    "gateway.public.port": "443",
    "gateway.trusted": "proxy1, proxy2, proxy3",
    "gateway.trusted_hosts": "lb.example.com, lb, ftp.example.com, www.example.com"
  }
}
```

When you are finished, save the json file.
Pass the json file with tsm to set the configuration, and then apply settings by running the following commands:

```bash
tsm settings import -f /path/to/file.json
tsm pending-changes apply
```

For more information about using configKey schemes, see Configuration File Example.

### Deploy a Coordination Service Ensemble

The Coordination Service is built on Apache ZooKeeper, an open-source project, and coordinates activities on the server, guaranteeing a quorum in the event of a failure, and serving as the source of "truth" regarding the server topology, configuration, and state. The service is installed automatically on the initial Tableau Server node, but no additional instances are installed as you add additional nodes. Because the successful functioning of Tableau Server depends on a properly functioning Coordination Service, we recommend that for server installations of three or more nodes, you add additional instances of the Coordination Service by deploying a new Coordination Service ensemble. This provides redundancy and improved availability in the event that one instance of the Coordination Service has problems.

- Hardware requirements
- The Coordination Service Quorum
- Deploy a new Coordination Service ensemble

### Hardware requirements

The hardware you use for Tableau Server can have an effect on how well the Coordination Service runs. In particular:

- **Memory.** The Coordination Service maintains state information in memory. By design, the memory footprint is small, and is typically not a factor in overall server performance.

- **Disk speed.** Because the service stores state information on disk, it benefits from
fast disk speed on the individual node computers.

- **Connection speed** between nodes. The service communicates continuously between cluster nodes; a fast connection speeds between nodes helps with efficient synchronization.

Because the Coordination Service is I/O intensive, if you are running Tableau Server on computers that meet or just exceed the minimum hardware requirements, you may want to configure a Coordination Service ensemble that puts the service on nodes that are not being used for other server processes. This reduces the chance of delays due to I/O contention between server processes. For information on how to deploy an ensemble on dedicated Coordination Service-only nodes, see Configure Tableau Server for High Availability with Coordination Service-Only Nodes.

**The Coordination Service Quorum**

To ensure that the Coordination Service can work properly, the service requires a *quorum*—a minimum number of instances of the service. This means that the number of nodes in your installation impacts how many instances of the Coordination Service you want to configure in your ensemble.

**Number of Coordination Service instances to use**

The maximum number of Coordination Service instances you can have in an ensemble on Tableau Server depends on how many Tableau Server nodes you have in your deployment. Configure a Coordination Service ensemble based on these guidelines:

<table>
<thead>
<tr>
<th><strong>Total number of server nodes</strong></th>
<th><strong>Recommended number of Coordination Service nodes in ensemble (must be 1, 3, or 5)</strong></th>
<th><strong>Notes</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>1-2 nodes</td>
<td>1 node</td>
<td>This is the default and requires no changes unless you want to move the Coordination Service off your initial node and onto your additional node.</td>
</tr>
<tr>
<td>Total number of server nodes</td>
<td>Recommended number of Coordination Service nodes in ensemble (must be 1, 3, or 5)</td>
<td>Notes</td>
</tr>
<tr>
<td>------------------------------</td>
<td>---------------------------------------------------------------------------------</td>
<td>-------</td>
</tr>
<tr>
<td>3-4 nodes</td>
<td>3 nodes</td>
<td></td>
</tr>
<tr>
<td>5 or more nodes</td>
<td>5 nodes</td>
<td>Five is the maximum number of Coordination Service instances you can install.</td>
</tr>
</tbody>
</table>

**Deploy a new Coordination Service ensemble**

If you install a total of three or more nodes, you should also deploy a Coordination Service ensemble. If you do not, you will get a warning message every time you make changes to the server configuration or topology. You can ignore this message, but as a best practice you should deploy a multi-node Coordination Service ensemble.

When you install Tableau Server, a single instance of the Coordination Service is installed on the initial node. TSM and Tableau Server depend on the Coordination Service to function properly, so to provide redundancy and ensure availability on multi-node installations, configure additional instances of the Coordination Service by deploying a Coordination Service ensemble. Coordination Service ensembles are installed with one, three, or five instances of the Coordination Service. In a three-node installation of Tableau Server, the recommended number of Coordination Service instances is three, one on each node.

Do not attempt to deploy a Coordination Service ensemble if there are other changes pending. Discard or apply any pending changes before deploying a new Coordination Service ensemble.
The following steps illustrate how to deploy a new Coordination Service ensemble on an existing three-node Tableau Server cluster and clean up the old ensemble.

1. On the initial node, open a terminal session.

2. Type this command to sign in to Tableau Server as a TSM administrator:

   \texttt{tsm login -u <username>}

   You will be prompted for your password.

3. Type this command to stop Tableau Server:

   \texttt{tsm stop}

   Some TSM processes will continue to run, including the Administration Controller and Administration Agent.

4. Confirm there are no pending changes:

   \texttt{tsm pending-changes list}

   If there are pending changes, you need to either discard the changes or apply them. Applying pending changes will take some time:

   - Discard the changes

     \texttt{tsm pending-changes discard}

     or
Apply the changes:

```
tsm pending-changes apply
```

Wait until the command completes and you are returned to the system prompt.

5. Get the node IDs for each node in the cluster:

```
tsm topology list-nodes -v
```

6. Use the `tsm topology deploy-coordination-service` command to add a new Coordination Service ensemble by adding the Coordination Service to specified nodes. You must specify the node(s) that the Coordination Service should be added to. The command also makes the new ensemble the "production" ensemble (the ensemble in use).

For example, deploy the Coordination Service to all three nodes of a three-node cluster, where the nodes are node1, node2, and node3:

```
tsm topology deploy-coordination-service -n node1,node2,node3
```

7. Wait until the new Coordination Service ensemble is running and the server is ready for the next step.

**Important.** If you attempt to clean up the old Coordination Service ensemble before the server is in the proper state, you can put the server into an unrecoverable state and may need to completely reinstall Tableau.

a. Check the status of the server:

```
tsm status -v
```

If the deployment is not complete, you may see processes showing as running when they are not, and the Coordination Service showing a status of
"unavailable" while the service is synchronizing between nodes on the cluster. Tableau Server may show as being in an error state while this is happening. You may also get an error message: "Could not connect to TSM Controller at '<host>:8850'." This is normal when the server is returning to a valid state.

b. Check the status of the server periodically until you are prompted to sign in again.

c. When you are prompted, sign in to TSM and continue to check the server status until you see a status of "STOPPED" for each node. If the status of a node shows as "ERROR" you need to wait. When each node status is "STOPPED" you should also see the following services running:

On the initial node:

- Two instances of the Coordination Service on the initial node, both with a status of "running".
- The Administration Controller with a status of "running". (The Administration Controller is only installed on the initial node.)
- The Administration Agent with a status of "running".
- Additional services on the initial node, all with a status of "running": Service Manager, License Manager, Client File Service.

On the additional nodes:

- One or more instance of the Coordination Service on each additional node you specified when you deployed the new ensemble, all with a status of "running". If you are deploying a new ensemble to nodes that already had Coordination Service running, you will see two instances of the service.
- The Administration Agent on every node, with a status of "running".
If you do not see a status of "running" for all of the above, wait a few minutes and run the status command again.

**Note:** If there is a problem with an instance of the Coordination Service (if it shows as stopped for example), you can toggle back to your previous Coordination Service ensemble using the `tsm topology toggle-coordination-service` command. To do this, the rest of the services should be in the state described above, including Administration Controller and Agent. You can toggle back to the previous ensemble only if you have not run the `cleanup-coordination-service` command. Tableau Server cannot be running when you use this command.

8. Remove the old ensemble. This step is required. You cannot run Tableau Server with multiple Coordination Service ensembles configured.

```
tsm topology cleanup-coordination-service
```

Tableau Server must be stopped when you use this command, but some TSM services will be running (see above).

9. Start Tableau Server:

```
tsm start
```

**Configure Client File Service**

The Client File Service (CFS) manages most shared files in a multinode cluster. For example, authentication related certificates, keys, and files (OpenID, mutual SSL, SAML, and Kerberos), and customization files are managed by CFS.

A single instance of CFS is configured on the initial node when you first install Tableau Services Manager (TSM) and Tableau Server. You can configure additional instances on other nodes, using these limitations and recommendations:
Tableau Server on Linux Administrator Guide

- There must be at least one instance of CFS for any Tableau Server installation. You cannot remove all instances of CFS.
- We recommend you do not configure more than one instance on any one node. Additional instances would not provide any benefit and would be a waste of resources.
- We recommend you configure a CFS instance on each node where you deploy the Coordination Service.

This article explains how to add or remove an instance of CFS. You cannot configure CFS from the Web UI. You must use the CLI to add or remove CFS. The procedure is the same one you use for adding a process to a node.

Configure CFS on additional nodes

1. On the initial node, open a terminal session.
2. Find the node ID for the node you are updating:
   
   ```
   tsm topology list-nodes -v
   ```
3. Add CFS on the node by specifying the process and a single instance.
   
   For example, this command adds one instance of CFS to node1:
   
   ```
   tsm topology set-process -n node1 -pr clientfileservice -c 1
   ```
4. Apply the changes:
   
   ```
   tsm pending-changes apply
   ```

Repository Failover

In a Tableau Server installation, the repository (pgsql) database is one of the key required processes. The Tableau Server repository stores information about Tableau Server users, groups and group assignments, permissions, projects, data sources, and extract metadata and refresh information. Because it is critical to the server functioning, Tableau Server has a
built-in automatic "failover" for the repository when server is installed in a distributed environment that meets certain requirements.

**Automatic repository failover**

Automatic repository failover means that if there is a problem with the active Tableau Server repository, the server will automatically switch to using the passive repository. This does not happen immediately, to protect against momentary issues with the repository that don’t justify a switch, but if the repository is unavailable for more than five minutes, failover occurs.

For automatic repository failover to work, your Tableau Server installation needs:

- A minimum of three nodes
- Two instances of the repository installed

Optional but highly recommended:

- A multi-node Coordination Service ensemble deployed

With these conditions satisfied, repository failover will occur if the active repository becomes unavailable, either due to a problem with the process, or a problem with the node the process is running on. If the original repository becomes available again (if, for example, the node is restarted and all processes come up properly), it is made the passive repository, available for failover if necessary.

**Manual repository failover**

There may be reasons you want to shift back to the original repository after failover occurs. One reason would be if that instance of the repository is installed on a computer with more resources. To do this, use the `tsm topology repository-failover` command to manually switch back to the original repository. For more information, see `tsm topology failover-repository`.
Preferred active repository

When you configure Tableau Server you have the option to specify a node as the preferred active repository. When Tableau Server is configured for repository failover, the preferred active repository node is the one used for the active repository. This is an optional step, and if you do not specify a preferred active repository node, Tableau Server will select the active repository node on startup.

To configure the preferred active repository, use the `tsm configuration set` command to configure the `pgsql.preferred_host` option:

```
  tsm configuration set -k pgsql.preferred_host -v "<host_name>"
```

**Note:** The `host_name` is case-sensitive and must match the node name shown in the output of `tsm status -v`.

Configure a preferred active repository node if you want Tableau Server to select a specific node on startup. You might want to do this if you have a particular server you want to use for your active repository (a computer with more disk space or memory for example), or if you are using custom administrative views. Custom administrative views have embedded connection information that refers to the repository for which you created the views. For more information on connecting to the Tableau Server repository, see Collect Data with the Tableau Server Repository.

Recover from an Initial Node Failure

The first computer you install Tableau on, the "initial node," has some unique characteristics. Two processes run only on the initial node and cannot be moved to any other node except in a failure situation, the License service (License Manager) and TSM Controller (Administration Controller). Tableau Server includes a script that automates moving these two processes to one of your other existing nodes so you can get complete access back to TSM and keep Tableau Server running.
Two other processes are initially included on the initial node but can be added or moved to additional nodes, the CFS (Client File Service) and the Coordination Service. Depending on how your installation was configured with CFS and Coordination Service, you may also need to take steps to redeploy these.

If there is a problem with the initial node and you have redundant processes on your other nodes, Tableau Server can continue to run for up to 72 hours before the lack of the licensing service impacts other processes. Your users can continue to sign in and see and use their content after the initial node fails, but you will not be able to reconfigure Tableau Server because you won’t have access to the Administration Controller. This means you should make a point of moving the two unique processes to another of your running nodes as soon as possible. If your initial node fails for reasons that are recoverable in a relatively short amount of time (for example, a hardware failure you can correct), you should first attempt to bring the node back up without using the procedure below.

**Note:** The steps in this article require server downtime and can be disruptive, and should only be used in the event of a catastrophic failure of the initial node. If you are unable to get your initial node running again, use the following steps to move key TSM processes to another node in your cluster.

### General requirements

- If the initial node was running the only instance of the Client File Service (CFS), you need to add that process to another node. Tableau Server requires at least one instance of the CFS. For more information, see Configure Client File Service.
- As part of the process for setting up a multi-node Tableau Server installation you should have deployed a Coordination Service ensemble. The process below assumes there was a Coordination Ensemble deployed before there was a problem with the initial node. For more information about deploying a Coordination Service ensemble, see Deploy a Coordination Service Ensemble.
Note: This operation includes steps that you may need to perform using the TSM command line.

Move the TSM Controller and License Service to another node

If there is a problem with the initial node, the TSM Controller and the Licensing Service need to be started on another node. Follow these steps to use the provided move-tsm-controller script and get the Controller and Licensing Service working on another node.

1. On a node that is still working, run the Controller recovery script. At a terminal prompt on a working node, type the following command:

   ```
   sudo /opt/tableau/tableau_server/packages/scripts.<version_code>/move-tsm-controller -n <nodeID>
   ```

   where "nodeID" is the ID for the node you want the TSM Controller to run on. For example:

   ```
   sudo /opt/tableau/tableau_server/packages/scripts.10400.17.0802.1319/move-tsm-controller -n node2
   ```

2. Verify the Administration Controller is running on the node:

   ```
   tsm status -v
   ```

3. Stop Tableau Server:

   ```
   tsm stop
   ```

4. Add the License Service to the node:

   ```
   tsm topology set-process -pr licenseService -n <nodeID> -c 1
   ```

5. Remove the old License Service from the original node, where "nodeID" is the initial
node that has failed:

```
tsm topology set-process -pr licenseservice -n <nodeID> -c 0
```

6. If the initial node had been running the only instance of CFS, add CFS to this node:

```
tsm topology set-process -pr clientfileservice -n node2 -c 1
```

7. If the initial node had been running the Messaging Service, add the Messaging Service to this node:

```
tsm topology set-process -pr activemqserver -n node2 -c 1
```

8. (Optional) You can also add other processes that had been running on the initial node but are not running on this node. For instance, to add an cache server:

```
tsm topology set-process -pr cacheserver -n node2 -c 1
```

9. Apply the changes:

```
tsm pending-changes apply
```

The `pending-changes apply` command displays a prompt to let you know this will restart Tableau Server if the server is running. The prompt displays even if the server is stopped, but in that case there is no restart. You can suppress the prompt using the `--ignore-prompt` option, but this does not change the restart behavior. For more information, see `tsm pending-changes apply`.

10. Restart the TSM Administration Controller (as `tableau` system account):

```
sudo su -l tableau -c "systemctl --user restart tabad-mincontroller_0.service"
```
11. Apply pending changes (there may not appear to be any, but this step is required):

```bash
tsm pending-changes apply
```

12. Activate the Tableau Server license on the new Controller node:

```bash
tsm licenses activate -k <product-key>
```

13. Verify the license is properly activated:

```bash
tsm licenses list
```

14. If the initial node was running the Coordination Service, you need to deploy a new Coordination Service ensemble that does not include that node. If you have a three node cluster and the initial node was running the Coordination Service, you must deploy a new, single-instance Coordination Service ensemble on a different node and clean up the old ensemble. In this example, a single instance of the Coordination Service is being deployed to the second node:

```bash
tsm topology deploy-coordination-service -n <nodeID2>
```

Wait until the server is completely switched over to the new ensemble.

15. When the server has switched over to the new ensemble, clean up the old ensemble.
Important: Do not do this too soon. You must wait until the server has completely switched to the new ensemble before running the cleanup command or you can permanently break Tableau. For more information about deploying a Coordination Service ensemble, including detailed instructions for determining that the server is ready to clean up the old ensemble, see Deploy a Coordination Service Ensemble.

16. If the initial node was running a File Store instance, you need to remove that instance:

   tsm topology filestore decommission -n <nodeID> --delete-filestore

   Where nodeID is the initial node that has failed.

17. Apply pending changes, using the --ignore-warnings flag if the new Coordination Service ensemble you deployed above is a single node ensemble:

   tsm pending-changes apply --ignore-warnings

18. Remove the initial node, where nodeID is the initial node that has failed:

   tsm topology remove-nodes -n <nodeID>

19. Apply pending changes, using the --ignore-warnings flag if the new Coordination Service ensemble you deployed above is a single node ensemble:

   tsm pending-changes apply --ignore-warnings

20. Start Tableau Server:

   tsm start
At this point your server should start, and you will be able to use TSM to configure it. The next step is to replace your initial node so your cluster has the original number of nodes. How you do this depends on whether or not you want to reuse the node that failed. We recommend that you only reuse that node if you are able to identify the reason it failed, and take steps to keep the failure from recurring.

21. If you plan to reuse the original node, you first need to completely remove Tableau from it. Do this by running the `tableau-server-obliterate` script. For details on doing this, see Remove Tableau Server from Your Computer.

22. On a fresh computer, or on your original computer after completely removing Tableau, install Tableau using your original Setup program and a bootstrap file generated from the node that is now running the Administration Controller and Licensing Service. This creates an additional node you can configure as part of your cluster. For details on how to add the node, see Install and Configure Additional Nodes.

A best practice is to configure any processes you lost when the original node failed, to make sure your cluster is fully redundant. You may want to move processes from your new initial node to the newly added additional node to duplicate your original configuration. For example, if your initial node was only running gateway and File Store, you may want to configure the new initial node the same way.

23. You should also redeploy a new Coordination Service ensemble, once you have your nodes up and running the way you want. For details, see Deploy a Coordination Service Ensemble.

**Recover from a Node Failure**

If there is a problem with one of your server nodes, and you have redundant processes on your other nodes, Tableau Server can continue to run. Your users can continue to sign in and see and use their content after the node fails, but they may experience performance degradation as a result of the failed node. In addition, your server will be at greater risk of
catastrophic failure if the bad node was running processes that are no longer redundant. This means you should make a point of removing the bad node and replacing it as soon as you can. If your node fails for reasons that are recoverable in a relatively short amount of time (for example, a hardware failure you can correct), you should first attempt to bring the node back up without using the procedure below.

**Note:** If the failed node is your *initial* node, there are larger implications for your Tableau Server installations. For details on how to recover from the failure of an initial node, see Recover from an Initial Node Failure.

**General requirements**

- There is still at least one functioning node with an instance of the File Store on it.
- There is still at least one functioning node with a Repository on it.

**Note:** This operation includes steps that you may need to perform using the TSM command line.

**Removing a Failed Node**

To remove a failed node from a Tableau Server cluster:

1. Stop Tableau Server:
   
   tsm stop

2. Identify the failed node:
   
   tsm status -v
The failed node will have a status of "ERROR" and processes will show as unavailable. The node ID is listed as "node<n>" with the machine name following it. For example, node3:

node3: WIN-00915SFASVH
   Status: ERROR
   'Tableau Server Gateway 0' status is unavailable.

3. Determine any key processes that were running on the node:

   - If the failed node was running the Messaging Service, you need to remove the service from the failed node and add it to a new node.

   Remove it from the failed node:

   ```
   tsm topology set-process -pr activemqserver -n <nodeID> -c 0
   ```

   Add it to a new node:

   ```
   tsm topology set-process -pr activemqserver -n <nodeID> -c 1
   ```

   - If the failed node was running the Coordination Service, you need to deploy a new ensemble and clean up the old one before you can remove the node. Do this while Tableau Server is stopped. For detail steps, see Deploy a Coordination Service Ensemble.

   - If the failed node was running File Store, you need to force-decommission File Store and remove it before you can remove the node.

   ```
   tsm topology filestore decommission -n <nodeID> --delete-filestore
   ```

   Apply pending changes (use `--ignore-warnings` option if you had a three
node cluster and a single Coordination Service instance):

tsm pending-changes apply --ignore-warnings

4. If the cluster was a three-node cluster and there are repositories on the remaining working nodes, you need to either remove one repository, or add a new node. This is because you are limited to a single instance of the repository when you have fewer than three nodes.

To remove one repository:

tsm topology set-process -n <nodeID> -pr psql -c 0

5. Run the command to remove the failed node. This adds the change to the pending changes list:

   tsm topology remove-nodes -n <nodeID>

6. Verify the node removal is pending:

   tsm pending-changes list

7. Apply pending changes to remove the node:

   tsm pending-changes apply

8. Start Tableau Server:

   tsm start

9. Install Tableau Server on a new node and configure the node with the processes that the old, failed node had been running.

10. On a fresh computer, or on your original computer after completely removing Tableau, install Tableau using your original Setup program and a bootstrap file generated from the initial node. For details on how to do this, see Install and Configure Additional Nodes.
A best practice is to configure any processes you lost when the original node failed, to make sure your cluster is fully redundant.

11. You should also redeploy a new Coordination Service ensemble, once you have your nodes up and running the way you want. For details, see Deploy a Coordination Service Ensemble.

Configure Nodes

Use the Tableau Services Manager (TSM) Web UI or CLI commands to configure the topology of a node. The initial node is configured with a default that includes all the processes used by TSM and Tableau Server. When you add additional nodes you need to specify which processes will run on those nodes, and how many instances of those processes will run. You may also want to change the topology of the initial node, either adding instances of existing processes, or moving some of those processes to your additional nodes (this is common when setting up a distributed installation of Tableau Server).

**Note:** You cannot remove the File Store or Repository (pgsql) if this is the only instance of that process in your cluster. You must have at least one instance of each of the processes in your Tableau Server installation, and you must add the second instance and allow it to synchronize with the first before you remove the process on the original node. For more information see Move the Repository Process and Move the File Store Process.

Use the TSM web interface

In most cases, you can make multiple changes to your server configuration. Exceptions are if you are moving or removing a File Store instance or the Repository.

To configure your Tableau Server nodes using the TSM web interface, do the following:
1. Open TSM in a browser:

https://<tsm-computer-name>:8850

For more information, see Sign in to Tableau Services Manager Web UI.

2. Click the **Configuration** tab.

Your next steps depend on the configuration changes you want to make.

- **Add processes**—Add processes to a node by specifying the number of instances you want on the node, or selecting the box for the process.

For example, to add four instances of the Backgrounder and a Gateway to a node:

   a. Select the **Gateway** box:

   ![Gateway Box]

   b. Set the **Backgrounder** count to 4:

   ![Backgrounder Count]

   ![Data Server Count]

   ![Data Engine Count]

   ![File Store Count]

   ![Repository Count]
Adding Backgrounder to a node will also add an instance of Data Engine if one is not already on the node.

**Note:** The TSM Web UI limits you to a maximum of 8 instances of processes that allow you to select the number of instances. To configure more instances than this, use the command line and the `TSM topology set-process` command. For more information, see `tsm topology set-process`.

- **Change process count**—Change the number of processes on a node by specifying the new number, or selecting the box for the process.

- **Remove a process completely**—Remove all instances of a process from a node by clearing the box for the process, or setting the count to 0 (zero).

In most cases you move a process from one node to another by setting the process instance count on the first node to 0 (zero), and setting the count to a non-zero value on the second node.

If you are attempting to make a configuration that is not allowed (if, for example, you try to remove a File Store that has not been decommissioned), a message displays to let you know this.

3. Click **Pending Changes** at the top right, and **Apply Changes and Restart** to commit the changes and restart Tableau Server.

**Use the TSM CLI**

- Adding processes to a node

- Changing the number of processes on a node
• Removing all instances of a process from a node

• Moving all instances of a process from one node to another node

To configure nodes, run commands from the initial node and use the node ID to specify which node you are configuring. To determine the node ID, use the `tsm topology list-nodes` command. Use the `tsm topology set-process` command to add, update or remove a process on a node. You need to specify the node you are configuring, the process you are adding, updating, or removing, and the number of instances of the process. After setting the topology for a node you need to apply the changes to Tableau Server.

Apply changes using the `tsm pending-changes apply` command. After the changes are applied, Tableau Server is returned to the state it was in before the command was run. This means that if it was running, it will be restarted, and if it was stopped it will remain stopped after pending changes have been applied. In most cases, if Tableau Server is running when you apply pending changes, the server is stopped so that changes can be applied, and then restarted. The exception is if you are changing the number of instances of Backgrounder, or VizQL Server on an existing node. With changes to those processes on an existing node, Tableau Server does not have to be stopped if it is running.

You need the node ID for a node in order to configure the node. To determine the node ID, use this command:

`tsm topology list-nodes -v`

**Note:** Examples here show some process names. For a complete list, see Tableau Server Processes.

Adding processes to a node

Use the `tsm topology set-process` command to add a process to a node. You need to specify the node you are configuring, the process you are adding, and the number of instances of the process.
1. On the initial node, open a terminal session.

2. Find the node ID for the node you are changing:
   
   tsm topology list-nodes -v

3. Add processes on the node by specifying the process and the number of instances.
   
   For example, this command adds two instances of backgrounder to node1:
   
   tsm topology set-process -n node1 -pr backgrounder -c 2

4. Apply the changes:
   
   tsm pending-changes apply

Changing the number of processes on a node

Change the number of processes on a node by specifying an already configured process and providing a new value for the number of instances.

1. On the initial node, open a terminal session.

2. Find the node ID for the node you are changing:
   
   tsm topology list-nodes -v

3. Change the number of processes on the node by specifying an already configured process and providing a new value for the number of instances.
   
   For example, on a node (node1) that is already running backgrounder, this command changes the number of instances to four:
   
   tsm topology set-process -n node1 -pr backgrounder -c 4

4. Apply the changes:
   
   tsm pending-changes apply
Removing all instances of a process from a node

1. On the initial node, open a terminal session.

2. Find the node ID for the node you are changing:
   
   tsm topology list-nodes -v

3. Remove a process from a node by specifying a count of 0 instances for that process on the node.

   For example, this command removes the backgrounder process from node1:
   
   tsm topology set-process -n node1 -pr backgrounder -c 0

4. Apply the changes:
   
   tsm pending-changes apply

Moving all instances of a process from one node to another node

In most cases you move a process from one node to another by setting the process instance count on the first node to zero, and setting the count to a non-zero value on the second node.

1. On the initial node, open a terminal session.

2. Find the node ID for the node you are changing:
   
   tsm topology list-nodes -v

3. Move a process from one node to another node by specifying a count of 0 instances for that process on the first node and specifying a count of 1 or greater to the second node.

   For example, these commands remove Backgrounder from node0 and add two instances of it to node1:
   
   tsm topology set-process -n node0 -pr backgrounder -c 0
Workload Management through Node Roles

Using node roles, you can configure where certain types of workloads are processed on your Tableau Server installation. The node roles feature allows you to dedicate and scale resources to specific workloads. You can configure node roles for Backgrounder and File Store.

Backgrounder Node Roles

The Backgrounder process runs Tableau Server tasks, including extract refreshes, subscriptions, flow tasks, ‘Run Now’ tasks, and tasks initiated from tabcmd. Running all these tasks can use a lot of machine resources. If you have more than one Backgrounder node in your cluster, you can manage your Backgrounder workload by specifying the type of tasks a Backgrounder can run on a node using the Backgrounder node role feature.

This configuration option is currently available only through TSM CLI commands and is only useful on multi-node clusters. If you have only one node, the Backgrounder is set to run all tasks by default and that cannot be changed.

Using Backgrounder Node Roles

The Backgrounder node role feature is intended to give you more control and governance over where certain type of Backgrounder workloads are processed in your Tableau Server installation and allows you to dedicate and scale resources to specific workloads.

For example, if your deployment is heavy on extract and users are running a lot of extract refreshes or encryption jobs, it could be beneficial to dedicate a node to extract refreshes. Similarly, in the case of subscriptions, if your Tableau Server installation processes a lot of subscriptions and you want to ensure that other jobs do not take resources from subscriptions, then you can dedicate a node to subscriptions. In these cases, you would also
want to dedicate other backgrounder nodes to workloads other than extract refreshes or subscriptions.

To support high availability, Tableau recommends having multiple nodes that are dedicated towards a specific workload. For example, if you dedicate a node to extract refreshes, you should also configure a second node to process extract refresh workload. This way if a node dedicated to extract refreshes becomes unavailable, extract refreshes can still be processed by the other node.

Configuration Options

<table>
<thead>
<tr>
<th>Configuration</th>
<th>Jobs</th>
</tr>
</thead>
<tbody>
<tr>
<td>all-jobs (default)</td>
<td>All Tableau Server jobs</td>
</tr>
<tr>
<td>flows</td>
<td>Flow run jobs.</td>
</tr>
<tr>
<td>no-docs</td>
<td>All jobs except flows.</td>
</tr>
<tr>
<td>extract-refreshes</td>
<td>Jobs that are created for:</td>
</tr>
<tr>
<td></td>
<td>Incremental refreshes, full refreshes, encryption and decryption of</td>
</tr>
<tr>
<td></td>
<td>all extracts including extracts that flow outputs create.</td>
</tr>
<tr>
<td>subscriptions</td>
<td>Subscription jobs</td>
</tr>
<tr>
<td>extract-refreshes-and-subscriptions</td>
<td>Extract-refreshes, encryption and decryption of all extracts including extracts that flow outputs generate, and subscription jobs.</td>
</tr>
<tr>
<td>no-extract-refreshes</td>
<td>All jobs except extract-refreshes, extract encryption and decryption of all extracts including extracts created from flow outputs.</td>
</tr>
<tr>
<td>no-subscriptions</td>
<td>All jobs except subscriptions.</td>
</tr>
<tr>
<td>no-extract-refreshes-and-subscriptions</td>
<td>All jobs except extract-refreshes, encryption and decryption of all extracts including extracts created from flow outputs, and subscriptions.</td>
</tr>
</tbody>
</table>
For more information on how to use the tsm commands to set the node role, see tsm topology.

**Note:** Making configurations to node roles require a restart of the server and will require some downtime. For more information, see tsm pending-changes.

License Requirements

Configuring a node to do only a specific type of tasks, like, flows, extract refreshes, and subscriptions, you must have one of the following licenses activated on your Tableau Server:

- To configure a node to run flows, you must have a valid Data Management Add-on license activated on your server, and have Tableau Prep Conductor running on that node. To learn more about Tableau Prep Conductor, see Tableau Prep Conductor.

- To configure a node to run all jobs except flows, you must have Tableau Prep Conductor running on the node. Tableau Prep Conductors requires a valid Tableau Data Management Add-on license activated on your server. For more information, see Licensing Tableau Prep Conductor.

- To configure a node to run extract refreshes, subscriptions, and any combination related to extract refreshes and subscriptions you must have a valid Server Management Add-on license activated on your Tableau Server. If the license expires or is deactivated, you will see an error any time you make a change to the Server configuration. For more information on Server Management Add-on. see About Tableau Server Management Add-on.

**Important!**

While flows, extract refreshes, and subscriptions can be expensive and resource heavy, they are not the only jobs that may require dedicated resources. In the all jobs group, there are a variety of System jobs that the Backgrounder executes, such as thumbnail generation for workbooks. Make sure that the nodes that run jobs other than extract refreshes, subscriptions, or flows have enough machine resources.
For more information on configuring node roles using TSM commands, see tsm topology set-node-role.

Considerations

There are some rules you must consider when configuring Backgrounder node roles, which are listed below:

- Only one node role configuration can be set for a node at a time. You cannot configure multiple node roles on a node.

- To configure a node role, there must be at least one Backgrounder process on that node.

- If you have only one Backgrounder node, you must configure this node to run all jobs. This is the default configuration and does not require additional licensing.

- If you have more than one Backgrounder node, combined, they must be configured to handle all jobs. This can be achieved in the following ways:
  
  - Configure one of the nodes to run all jobs using the all jobs option. This is the easiest and most straightforward way.

  - Using one of the exception configurations on one of the nodes:
    - no-flows
    - no-subscriptions
    - no-extract-refreshes
    - no-extract-refreshes-and-subscriptions

For example, in a cluster where there are three backgrounders, you could have one node configured to run flows, one to run subscriptions and extract refreshes, and one to run all jobs except flows, subscription and extract refreshes.

**Note:** The ability to specify node roles to run flows, or run all jobs except flows, or run all jobs was introduced in 2019.1.
File Store Node Roles

The Tableau Server File Store controls the storage of extracts. There are three broad categories of workloads that are extract dependent.

<table>
<thead>
<tr>
<th>Extract Workload</th>
<th>Execution Service</th>
</tr>
</thead>
<tbody>
<tr>
<td>Refresh</td>
<td>Backgrounder</td>
</tr>
<tr>
<td>Query</td>
<td>Data Engine</td>
</tr>
<tr>
<td>Backup/Restore</td>
<td>Backup/Restore</td>
</tr>
</tbody>
</table>

File Store node role management in combination with Backgrounder node role management gives server admins the ability to specialize server nodes to preferentially execute selected workloads to optimize performance of all categories of extract heavy workloads.

It is possible to specialize a node to execute extract query workloads through a topology that has only stand-alone Data Engine nodes. For more information, see Optimize for Query Heavy Environments. However, this is at the expense of extract refresh workloads, which are executed by Backgrounder nodes. With the topology-based isolation approach, extract refresh heavy Backgrounder workloads can get slower as none of the Backgrounder nodes have a File Store and thus all extract refresh traffic goes over the network.

With the File Store Node Role configuration option, it is possible to designate certain server nodes that process extract queries to be preferentially selected from the list of server nodes that can do so. This helps speed up workloads such as backup and extract refreshes by allowing server admins to enable File Store on Backgrounder server nodes, which prevents extract queries from running on these nodes. This feature is useful if you have an extract-heavy query workload and an extract-heavy refresh workload and want to achieve optimal extract query and refresh performance.

Here are guidelines to optimize for extract-refresh and backup/restore workloads.
Start from a topology with specialized Data Engine nodes (see Optimize for Query Heavy Environments).

<table>
<thead>
<tr>
<th>Process</th>
<th>Initial Node</th>
<th>Additional Node 1</th>
<th>Additional Node 2</th>
<th>Additional Node 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cluster Controller</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>Gateway</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>Application Server</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>VizQL Server</td>
<td>✔️ ✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>Cache Server</td>
<td>✔️ ✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>Search &amp; Browse</td>
<td>✔️ ✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>Backgrounder</td>
<td>✔️ ✔️ ✔️ ✔️</td>
<td>✔️ ✔️ ✔️ ✔️ ✔️</td>
<td>✔️ ✔️ ✔️ ✔️ ✔️</td>
<td>✔️ ✔️ ✔️ ✔️ ✔️</td>
</tr>
<tr>
<td>Data Server</td>
<td>✔️ ✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>Data Engine</td>
<td>✔️ ✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>File Store</td>
<td>✔️ ✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td>Repository</td>
<td>✔️ ✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
</tbody>
</table>

Topology 1 - Dedicated Data Engine Nodes

1. Add File Store to Node 0.
   
   tsm topology set-process -n node0 -pr filestore -c 1

2. Designate Node 2 and Node 3 to preferentially execute extract-query workloads.
   
   tsm topology set-node-role -n node2, node3 -r extract-queries

3. Designate Node 0 to preferentially execute extract-refresh workloads.
   
   tsm topology set-node-role -n node0 -r extract-refreshes

4. Designate Node 1 to preferentially execute non-extract-refresh workloads.
   
   tsm topology set-node-role -n node0 -r no-extract-refreshes

5. Apply pending changes.
   
   tsm pending-changes apply
Note: In your Tableau Server deployment, adding File Store roles to existing nodes will temporarily increase network I/O between all File Store nodes while the new File Store is being synchronized. The duration of this operation is dependent on the volume of data on the File Store and the network bandwidth capacity. The status of synchronization can be monitored using the TSM Web GUI. If you are adding more than one File Store to your deployment, it is recommended to add them consecutively and wait for the initial synchronization to complete in between each File Store addition.

Configuration Options

<table>
<thead>
<tr>
<th>Configuration</th>
<th>Jobs</th>
</tr>
</thead>
<tbody>
<tr>
<td>all-jobs (default)</td>
<td>All Tableau Server jobs</td>
</tr>
<tr>
<td>extract-queries</td>
<td>Jobs that are created for extract queries. The nodes selected will run as all-jobs and will prioritize the processing of extract queries.</td>
</tr>
</tbody>
</table>

For more information on configuring node roles using TSM commands, see tsm topology set-node-role.
License Requirements

To configure a node to run extract queries you must have a valid Server Management Add-on license activated on your Tableau Server.

Install Tableau Server on a Two-Node Cluster

When you install Tableau Server on a two-node cluster, you can install server processes on one or both nodes. A two-node cluster can improve the performance of Tableau Server, because the work is spread across multiple machines.

Note the following about two-node clusters:

- A two-node cluster does not provide failover or support for high availability.
- You can’t install more than one instance of the repository on a two-node cluster, and the repository must be on the initial node.

If you need failover or high availability, or want a second instance of the repository, you must install Tableau Server on a cluster of at least three computers. In a cluster that includes at least three nodes, you can configure two instances of the repository, which gives your cluster failover capability.

Maintain a Distributed Environment

After you set up an initial node and one or more additional nodes for a distributed installation, you can perform all subsequent configuration and updates from the initial node, using the TSM CLI, or from any computer using a browser and the TSM Web interface.

When you install additional nodes, they are added by computer name. If the computer name of a node changes, you will need to remove and reinstall the node. For details on removing a node, see Remove a Node.

You can monitor the status of the Tableau Server cluster on the TSM Status page. See View Server Process Status for details.

Additional actions you may need to maintain your distributed environment include:
Move the Repository Process

Tableau Server relies on the PostgreSQL repository to store server data. There must always be at least one active instance of the repository in any Tableau Server installation, and you can have a maximum of two instances (one active, one passive) if you have at least three nodes in your installation. You cannot remove a repository instance if it is the only instance.

This means that if you want to move the only instance of your repository from one node to another node, you need to add a second instance and synchronize the new repository with the old one before you remove the old one. Synchronize repository instances by starting server. If you've added a new repository, it will automatically synchronize with the existing instance.

If you are deleting a node from your server cluster and that node is hosting the only instance of the repository, you must add a second instance of the repository and synchronize the instances before removing the node.

If you are also moving the file store, you can move the repository at the same time. See Move the File Store Process.

Before making a change to the repository, create a full backup of Tableau Server. For more information, see tsm maintenance backup.

Important: You cannot add a second repository instance and remove the first one in the same step. You must have both running so the contents of the first is synchronized with the second, before you remove the original instance.

The steps for moving the repository are:

1. Add a new instance of the repository to another node, start server, and wait for it to synchronize with the first repository.
2. Remove the instance of the repository from the original node.
Use the TSM web interface

To move a repository you need to first add a second instance on a second node, and then after the two instances have synchronized all the data in the original repository, remove the original instance. These steps must be done separately to allow for the synchronization of content between the two instances.

Add a new instance of the repository.

1. Open TSM in a browser:

   https://<tsm-computer-name>:8850

   For more information, see Sign in to Tableau Services Manager Web UI.

2. Click the Configuration tab.

3. For the node you are adding the repository to:

   Select Repository (pgsql).

4. Click Pending Changes at the top of the page:

   The Pending Changes list displays.

   If you are configuring a three- or five-node cluster and have not deployed a Coordination Service ensemble, a warning will display. You can continue, and deploy a Coordination Service ensemble in a separate step. For details on deploying a Coordination Service ensemble, see Deploy a Coordination Service Ensemble.

5. Click Apply Changes and Restart and Confirm to confirm a restart of Tableau Server.

6. After Tableau Server has restarted, on the Status tab, verify that all processes are active.
Remove an instance of the repository.

1. In TSM, on the **Status** tab, verify that all processes are active. When both repositories show as Active, you can remove the first one.

2. Click the **Configuration** tab.

3. For the node you’re removing the repository from, clear the **Repository** box.

4. Click **Pending Changes** at the top of the page.

   If you are configuring a three- or five-node cluster and have not deployed a Coordination Service ensemble, a warning will display. You can continue, and deploy a Coordination Service ensemble in a separate step. For details on deploying a Coordination Service ensemble, see Deploy a Coordination Service Ensemble.

5. Click **Apply Changes and Restart** and **Confirm** to confirm a restart of Tableau Server.

Use the TSM CLI

To move a repository you need to first add a second instance on a second node, and then after the two instances have synchronized all the data in the original repository, remove the original instance. These steps must be done separately to allow for the synchronization of content between the two instances.

Add a new instance of the repository.

1. Add the repository (pgsql) to another node:

   ```bash
   tsm topology set-process -n <nodeID> -pr psql -c 1
   ```

2. Apply the changes. The `pending-changes apply` command displays a prompt to let you know this will restart Tableau Server if the server is running. The prompt displays even if the server is stopped, but in that case there is no restart. You can suppress the prompt using the `--ignore-prompt` option, but this does not change the
restart behavior. For more information, see tsm pending-changes apply.

```bash
tsm pending-changes apply
```

3. Wait for the new repository on the second node to synchronize with the repository on the first node.

```bash
tsm status -v
```

Wait until the new repository status shows as "passive".

Remove an instance of the repository.

Once the new instance of the repository is fully synchronized and shows as "passive" you can remove the original instance:

1. Remove the repository from the first node by setting the process count to 0 (zero):

```bash
tsm topology set-process -n <nodeID> -pr psql -c 0
```

2. Apply the change. The `pending-changes apply` command displays a prompt to let you know this will restart Tableau Server if the server is running. The prompt displays even if the server is stopped, but in that case there is no restart. You can suppress the prompt using the `--ignore-prompt` option, but this does not change the restart behavior. For more information, see tsm pending-changes apply.

```bash
tsm pending-changes apply
```

**Move the File Store Process**

You cannot remove an instance of File Store if it is the only instance on the server. You cannot run Tableau Server without at least one instance of File Store. This means if you need to move the File Store, or if you are deleting a server node that is hosting the only instance of the File Store, you must first move File Store to another node.

Moving the File Store is a two-part process:
Adding a second instance of File Store (if there is not an existing second instance).

Decommissioning and removing the original instance of File Store.

This article assumes you have installed Tableau Server on an initial node and at least one additional node. For more information on adding nodes to Tableau Server, see Install and Configure Additional Nodes.

Use the TSM web interface

Adding a second instance of File Store

This procedure assumes you have added an additional node. For more information on adding nodes to Tableau Server, see Install and Configure Additional Nodes.

1. Open TSM in a browser:

   https://<tsm-computer-name>:8850

   For more information, see Sign in to Tableau Services Manager Web UI.

2. Click the Configuration tab.

3. For the node you want to add an instance of File Store on, select File Store.

   Adding File Store to a node will also add an instance of Data Engine if one is not already on the node.

4. Click Pending Changes at the top of the page:

   The Pending Changes list displays.

   A Coordination Service ensemble warning displays because you are configuring a three-node cluster. You can continue. You will deploy a Coordination Service ensemble in an upcoming step.
5. Click **Apply Changes and Restart** and **Confirm** to confirm a restart of Tableau Server.

Decommissioning and removing an instance of file store

1. In TSM, on the **Status** tab, verify that all processes are active.

2. Click the **Configuration** tab.

3. For the node you’re removing File Store from, clear the **File Store** box.

   This will generate a warning about decommissioning the File Store before removing it. You cannot remove an instance of File Store unless it has been decommissioned. Click **OK** to have TSM decommission the file store for you, or **Cancel** to decommission the File Store from the command line. For more information about decommissioning the File Store using the TSM CLI, see Decommissioning and removing an instance of File Store below.

4. Click the **Status** tab to see the status of the decommission. When the instance of File Store is marked "Ready for removal" you can continue.

   ![File Store Status](image)

5. Click **Pending Changes** at the top of the page.

   **Note:** If you are configuring a three- or five-node cluster, a Coordination Service ensemble warning will display. You can continue to apply pending changes and deploy a Coordination Service ensemble in a separate step. For information on deploying a Coordination Service ensemble, see Deploy a Coordination Service Ensemble .

6. Click **Apply Changes and Restart** and **Confirm** to confirm a restart of Tableau Server.
Use the TSM CLI

Adding a second instance of file store

1. Create a full backup of Tableau Server. For more information, see Back up Tableau Server data.

2. Add the File Store to a second node.

   tsm topology set-process -n <nodeID> -pr filestore -c 1

   The File Store is automatically added. Data engine is also added if it is not already on the node.

   Apply the configuration changes:

   tsm pending-changes apply

   The pending-changes apply command displays a prompt to let you know this will restart Tableau Server if the server is running. The prompt displays even if the server is stopped, but in that case there is no restart. You can suppress the prompt using the --ignore-prompt option, but this does not change the restart behavior. For more information, see tsm pending-changes apply.

3. Check the status of the new file store instance.

   tsm status -v

   Wait for the new File Store to synchronize with the File Store on the first node. When synchronization is complete the new File Store has a status of "running" instead of "synchronizing".

Decommissioning and removing an instance of File Store

Once you have a second instance of File Store installed and synchronized you can decommission and remove the original instance. You must decommission the original instance before you remove it. Doing this guarantees that any unique files on the File Store node are duplicated to another file store node.
1. Decommission the original File Store:

   tsm topology filestore decommission -n <nodeID> --override

2. When the decommission command completes, remove the File Store from the node by applying the pending configuration changes. The File Store is automatically removed. Data Engine is also removed unless an instance of one of these processes is installed on the node: VizQL Server, Application Server (Vizportal), Data Server, or Backgrounder.

   tsm pending-changes apply

   The pending-changes apply command displays a prompt to let you know this will restart Tableau Server if the server is running. The prompt displays even if the server is stopped, but in that case there is no restart. You can suppress the prompt using the --ignore-prompt option, but this does not change the restart behavior. For more information, see tsm pending-changes apply.

Move the Messaging Service Process

This article assumes you have installed Tableau Server on an initial node and at least one additional node. For more information on adding nodes to Tableau Server, see Install and Configure Additional Nodes.

You cannot remove an instance of the Messaging Service if it is the only instance on the server. You cannot run Tableau Server without one instance of the Messaging Service. This means if you need to move the Messaging Service, or if you are deleting a server node that is hosting the Messaging Service, you must first move the service to another node.

Moving the Messaging Service is straightforward process of three steps:

- Remove the original instance of the Messaging Service.

- Add a new instance of the Messaging Service.

- Apply the pending changes.

Use the TSM web interface
Moving the Messaging Service

1. In TSM, on the **Status** tab, verify that all processes are active.

2. Click the **Configuration** tab.

3. For the node you’re removing the Messaging Service from, clear the **Messaging Service** box.

   This will activate the **Pending Changes** button, but in Pending Changes an error will tell you that the Messaging Service (activemqserver) is not on any node. Until you add it to another node, you cannot apply pending changes.

4. For the node you are adding the Messaging Service to, select the **Messaging Service** box.

5. Click **Pending Changes** at the top of the page.

6. Click **Apply Changes and Restart** and **Confirm** to confirm a restart of Tableau Server.

Use the TSM CLI

Moving the Messaging Service

1. On the initial node, open a terminal session.

2. Find the node ID for the nodes you are changing:

   ```
   tsm topology list-nodes -v
   ```

3. Remove the Messaging Service from one node:

   ```
   tsm topology set-process -n <nodeID> -pr activemqserver -c 0
   ```

4. Add the Messaging Service to another node:
5. Apply the configuration changes:

```bash
tsm topology set-process -n <nodeID> -pr activemqserver -c 1
```

The `pending-changes apply` command displays a prompt to let you know this will restart Tableau Server if the server is running. The prompt displays even if the server is stopped, but in that case there is no restart. You can suppress the prompt using the `--ignore-prompt` option, but this does not change the restart behavior. For more information, see `tsm pending-changes apply`.

Remove a Node

If your Tableau Server installation includes a node you no longer need, you can remove it to simplify your installation, and to free up the hardware resources on that node.

Prerequisites for removing a node

There are prerequisites you must satisfy before you can remove a node from your Tableau Server cluster. If your node has one of a number of limitations, you need to address these before you can remove the node. The limitations include confirm that the node has been added with at least one process, and that the node does not include any process that is not also installed on another node.

If one of the following configuration limitations applies, you must take action before you can remove the node:

- If the node was just added using the Web UI, you need to apply pending changes before you can remove it. If you added it using the command line, you need to configure it with at least one process before you can remove it.

- If the node includes the only Repository instance, you need to move the Repository to another node. See Move the Repository Process.

- If the node is running the only instance of the File Store, you need to move the File
Store to another node. See Move the File Store Process.

- If the node is running an instance of the Coordination Service, you must deploy a new Coordination Service ensemble that does not include the node. See Deploy a Coordination Service Ensemble.

- If the node is running the Messaging Service, you need to move the Messaging Service to another node. See Move the Messaging Service Process.

**Important:** Do not use the `tableau-server-obliterate` script to remove a node. First remove the node using the TSM UI or the `tsm topology remove-nodes` command. This leaves the rest of the cluster in a good state. Later, if you want to re-add the node to the cluster, run the `tableau-server-obliterate` script on the node to completely remove Tableau. After removing Tableau from the node, restart the computer and then re-add the node using the normal steps for adding and configuring a node. For details on running the script, see Remove Tableau Server from Your Computer.

Use the TSM web interface

1. Open TSM in a browser:
   
   https://<tsm-computer-name>:8850

   For more information, see Sign in to Tableau Services Manager Web UI.

2. Click the **Configuration** tab and, under the node you want to remove, click **Remove Node**:
If a configuration limitation does not allow you to remove the node (if, for example, it includes a File Store that must first be decommissioned), a message will display letting you know this. See Prerequisites above for more information.

**Note:** If you just added the node, and have not configured it, you must first apply pending changes before you can remove the node.

3. Click Pending Changes at the top right, and Apply Changes and Restart.

Use the TSM CLI

Use the `tsm topology remove-nodes` command to remove a node from a cluster.

To remove a node from a cluster it must have been configured with a process at some point in the past. If you added a node using the CLI and want to remove it but have not configured any processes, you must add a process on it, run the `tsm pending-changes apply` command, and then remove the node. For example, you might add one instance of Cluster Controller to the node:

```
 tsm topology set-process -n <nodeID> -pr clustercontroller -c 1.
```

If you are removing a node you added using the Web UI, the Cluster Controller process is automatically added so you do not need to add it before removing the node.

1. On the initial node, open a terminal session.

2. Find the node ID for the node you are changing:

```
 tsm topology list-nodes -v
```

3. Remove a node using the `remove-nodes` command.

   For example, to remove `node2` from an existing cluster:

```
 tsm topology remove-nodes --node-names "node2"
```
Configure Tableau Server for High Availability with Coordination Service-Only Nodes

The Coordination Service is built on Apache ZooKeeper, an open-source project, and coordinates activities on the server, guaranteeing a quorum in the event of a failure, and serving as the source of “truth” regarding the server topology, configuration, and state. The service is installed automatically on the initial Tableau Server node, but no additional instances are installed as you add additional nodes. Because the successful functioning of Tableau Server depends on a properly functioning Coordination Service, we recommend that for server installations of three or more nodes, you add additional instances of the Coordination Service by deploying a new Coordination Service ensemble. This provides redundancy and improved availability in the event that one instance of the Coordination Service has problems.

The Coordination Service can generate a large amount of I/O as it communicates with other components of the server, so if you are running Tableau Server on computers that meet or just exceed the minimum hardware requirements, you may want to install Tableau Server in a configuration that uses Coordination Service-only nodes. This means installing Coordination Service on nodes that run no other server processes, and removing Coordination Service from any nodes that are running other server processes. This procedure explains how to do this. You can also run the Coordination Service ensemble on the same nodes running other Tableau Server processes. For details on how to do that, see Deploy a Coordination Service Ensemble.

Prerequisite

Before proceeding with the procedures in this topic, complete the following prerequisites:
• Install and Configure Tableau Server - Install Tableau on your initial node.

• Install and Configure Additional Nodes - Install Tableau on at least two additional nodes.

**Note:** This operation includes steps that you may need to perform using the TSM command line.

Deploy an ensemble on Coordination Service-only nodes

One way to accommodate the high I/O impact of the Coordination Service is to deploy an ensemble on nodes that only run the Coordination Service and the Cluster Controller. The following steps illustrate how to deploy a Coordination Service ensemble on an existing multi-node Tableau Server cluster.

**Note:** For a core-based Tableau Server license, Coordination Service-only nodes do not count against the total count of licensed cores.

1. Add additional nodes to your cluster.

   See Install and Configure Additional Nodes.

2. If you added the new nodes using the TSM CLI, you need to configure the nodes with Cluster Controller (this step is not necessary if you added the nodes using the TSM Web UI because Cluster Controller is automatically added when you add a node with the Web UI).

   On the initial node, open a terminal session.

3. Type this command to sign in to Tableau Server as a TSM administrator:

   ```
   tsm login -u <username>
   ```

   You will be prompted for your password.
4. From the initial node of the cluster, configure the new nodes with an instance of the Cluster Controller:

   tsm topology set-process -pr clustercontroller -n <node4> -c 1
   tsm topology set-process -pr clustercontroller -n <node5> -c 1
   tsm topology set-process -pr clustercontroller -n <node6> -c 1

5. Apply the configuration changes. The `pending-changes apply` command displays a prompt to let you know this will restart Tableau Server if the server is running. The prompt displays even if the server is stopped, but in that case there is no restart. You can suppress the prompt using the `--ignore-prompt` option, but this does not change the restart behavior. For more information, see `tsm pending-changes apply`.

   tsm pending-changes apply

   A warning about deploying a Coordination Service ensemble displays because you have deployed a multi-node cluster. If this is the only warning, you can safely override it using the `--ignore-warnings` option to apply the configuration changes in spite of the warning.

   tsm pending-changes apply --ignore-warnings

6. Confirm that all nodes are up and running:

   tsm status -v

7. On the initial node of the cluster, open a terminal session and type this command to stop Tableau Server:

   tsm stop

8. Get the node IDs for each node in the cluster:
tsm topology list-nodes -v

9. Use the `tsm topology deploy-coordination-service` command to add a new Coordination Service ensemble by adding the Coordination Service to specified nodes. You must specify the node(s) that the Coordination Service should be added to. The command also switches Tableau Server to use the new ensemble.

For example, deploy the Coordination Service to three nodes of a six-node cluster:

```bash
tsm topology deploy-coordination-service -n <node4,node5,node6>
```

10. Wait until the new Coordination Service ensemble is running and the server is ready for the next step. **This is important.**

**Important.** If you attempt to clean up the old Coordination Service ensemble before the server is in the proper state, you can put the server into an unrecoverable state and may need to completely reinstall Tableau.

   a. Check the status of the server:

   ```bash
tsm status -v
   ```

   If the deployment is not complete, you may see processes showing as running when they are not, and the Coordination Service showing a status of "unavailable" while the service is synchronizing between nodes on the cluster. Tableau Server may show as being in an error state while this is happening. You may also get an error message: "Could not connect to TSM Controller at '<host>:8850'." This is normal when the server is returning to a valid state.

   b. Check the status of the server periodically until you are prompted to sign in again.

   c. When you are prompted, sign in to TSM and continue to check the server
status until you see a status of "STOPPED" for each node. If the status of a node shows as "ERROR" you need to wait. When each node status is "STOPPED" you should also see the following services running:

On the initial node:

- Two instances of the Coordination Service on the initial node, both with a status of "running".
- The Administration Controller with a status of "running". (The Administration Controller is only installed on the initial node.)
- The Administration Agent with a status of "running".
- Additional services on the initial node, all with a status of "running": Service Manager, License Manager, Client File Service.

On the additional nodes:

- One or more instance of the Coordination Service on each additional node you specified when you deployed the new ensemble, all with a status of "running". If you are deploying a new ensemble to nodes that already had Coordination Service running, you will see two instances of the service.
- The Administration Agent on every node, with a status of "running".

If you do not see a status of "running" for all of the above, wait a few minutes and run the status command again.

**Note:** If there is a problem with an instance of the Coordination Service (if it shows as stopped for example), you can toggle back to your previous Coordination Service ensemble using the `tsm topology toggle-coordination-service` command. To do this, the rest of the services
should be in the state described above, including Administration Controller and Agent. You can toggle back to the previous ensemble only if you have not run the `cleanup-coordination-service` command. Tableau Server cannot be running when you use this command.

11. When the new ensemble is running properly, remove the old ensemble. This step is required. You cannot run Tableau Server with multiple Coordination Service ensembles configured.

    `tsm topology cleanup-coordination-service`

    Tableau Server must be stopped when you use this command.

12. Start Tableau Server:

    `tsm start`

Add a Load Balancer

You can enhance the reliability of Tableau Server by running gateways on multiple nodes, and configuring a load balancer to distribute requests across the gateways. Unlike the repository process, which can be active or passive, all gateway processes are active. If one gateway in a cluster becomes unavailable, the load balancer stops sending requests to it. The load balancer algorithm you choose determines how the gateways will route client requests.

Notes: If you will be using Kerberos authentication, you need to configure Tableau Server for your load balancer before you configure Tableau Server for Kerberos. For more information, see Configure Kerberos.

    Windows SSPI authentication is not supported with a load balancer.

Note the following as you configure your load balancer to work with Tableau Server:
Tested load balancers: Tableau Server clusters with multiple gateways have been tested with Apache and F5 load balancers.

If you are using an Apache load balancer and creating custom administrative views, you need to connect directly to the Tableau Server repository. You cannot connect through the load balancer.

Tableau Server URL: When a load balancer is in front of a Tableau Server cluster, the URL that's accessed by Tableau Server users belongs to the load balancer, not the initial Tableau Server node.

Trusted host settings: The computer running the load balancer must be identified to Tableau Server as a trusted host. See the procedure below for how to configure Tableau Server.

Proxy server configurations: The settings used to identify a load balancer to Tableau Server are the same ones that are used to identify a proxy server. If your Tableau Server cluster requires both a proxy server and a load balancer, both must use a single external URL defined in `gateway.public.host` and all proxy servers and load balancers must be specified in `gateway.trusted` and `gateway.trusted_hosts`. For more information, see Configuring Proxies for Tableau Server.

Configure Tableau Server to work with a load balancer

Use the following key-value pairs to configure Tableau Server for a load balancer. You can set the values individually with the `tsm configuration set` command.

Or, use the configuration file template example at the end of this topic to create a configKey json file. After you have filled in the options with the appropriate values, pass the json file and apply settings with tsm commands.

Load balancer configuration key-value pairs

The key-value pairs described in this section can be set with the `tsm configuration set` command. After you have set all values, run the following command:
tsm pending-changes apply

gateway.public.host

Value: name

The value, name is the URL that will be used to reach Tableau Server through the load balancer.

For example, if Tableau Server is reached by entering tableau.example.com in a browser address bar, enter this command:

    tsm configuration set -k gateway.public.host -v "tableau.example.com"

gateway.public.port

Value: port_number

By default, Tableau assumes that the load balancer is listening on port 80 for external communications. Use this key to specify a different port.

For example, if your load balancer is configured for SSL and listening on port 443, enter the following command:

    tsm configuration set -k gateway.public.port -v "443"

gateway.trusted

Value: server

Specify the load balancer(s) IPv4 address or computer name.

The value for server can be a comma-separated list, for example:

    tsm configuration set -k gateway.trusted -v "10.32.139.45, 10.32.139.46, 10.32.139.47"

or
Tableau Server on Linux Administrator Guide

tsm configuration set -k gateway.trusted -v "proxy1, proxy2, proxy3"

gateway.trusted_hosts

Value:

Specify alternate names for the load balancer(s), such as its fully-qualified domain name, any non-fully-qualified domain names, and any aliases. These are the names a user might type in a browser. Separate each name with a comma:

tsm configuration set -k gateway.trusted_hosts -v "name1, name2, name3"

For example:

tsm configuration set -k gateway.trusted_hosts -v "lb.example.com, lb, ftp.example.com, www.example.com"

Configuration template example

You can configure all of the settings in one operation by customizing the following configuration template example and creating a configKey.json file.

Refer to the key-value pair descriptions above to customize the values for your organization.

```
{
  "configKeys": {
    "gateway.public.host": "tableau.example.com",
    "gateway.public.port": "443",
    "gateway.trusted": "proxy1, proxy2, proxy3",
    "gateway.trusted_hosts": "lb.example.com, lb, ftp.example.com, www.example.com"
  }
}
```

When you are finished, save the json file.
Pass the json file with tsm to set the configuration, and then apply settings by running the following commands:

```bash
tsm settings import -f /path/to/file.json

 tsm pending-changes apply
```

For more information about using configKey schemes, see Configuration File Example.

## Migrate

The topics in this section provide information on migrating between hardware, operating system, and cloud platforms for Tableau Server on Linux.

Looking for Tableau Server on Windows? See Migrate

To migrate a site from one Tableau Server to another, see Export or Import a Site.

To copy or migrate content between Tableau Server projects using the Tableau Content Migration Tool, see About Tableau Content Migration Tool.

## Changing the Identity Store

Infrastructure or business changes may require you to change the identity store on Tableau Server. There are two kinds of identity stores: local and external. When you installed Tableau Server you configured either a local identity store or an external identity store.

When you configure Tableau Server with a local identity store, all user and group information is stored and managed in the Tableau Server repository. In the local identity store scenario, there is no external source for users and groups.

When you configure Tableau Server with an external store, all user and group information is stored and managed by an external directory service. Tableau Server must synchronize with the external identity store so that local copies of the users and groups exist in the
Tableau Server repository, but the external identity store is the master source for all user and group data. Examples of external identity stores are OpenLDAP and Active Directory.

For more information about the Tableau identity store, see Identity Store.

You can change from local store to an external store, or you can change from an external store to a local store. In either case, to change the identity store type, you complete these steps:

1. Uninstall (including manual deletion of directories) and then reinstall of Tableau Server. The procedure for full uninstall and clean install are at the end of this topic.

2. Restore content and permissions.

In these steps the term "restore" does not refer to using the TSM maintenance restore command to restore the backup you are making. You cannot restore a backup (.tsbak) created on a Tableau Server instance that uses a different identity store than the target Tableau Server. The backup is a best practice safeguard, in case you need to go back to your original Tableau Server configuration.

Warning

Changing the installation type on Tableau Server can be a complicated and time-consuming process. To avoid data loss or orphaning of content or users, you'll need to plan this process carefully. In all cases, user filters applied to workbooks and data sources will need to be updated manually after the change.

Most importantly, determine how you will transition content and permissions to the new identity store after you reinstall Tableau Server.

Methods for restoring content and permissions

The following list describes two methods for restoring content and permissions after you reinstall Tableau Server. Select the method that best fits with your environmental requirements.
• **Method 1: Use site export and import**—In this method, you start by exporting each site in your existing deployment. Then, you install the new server and configure it for the new identity store type. You then create new users in the default site on the new server. Finally, you import all the original sites. During the import stage, you can map the original identities to the new users that you created in the default site.

Because this method exports all content and permissions at each site, it is the best method for organizations that require a high fidelity replica of the content and permissions after the identity store change is complete. Some organizations require an identity store change as the result of an authentication change. In these cases, a different user name syntax is a often a requirement in the new model. This method, which includes a process of mapping original user names to new names, provides flexibility for such scenarios.

• **Method 2: Fresh installation; users republish content**—In this method, you install a new version of Tableau Server and select the new identity store type during setup. You also create new sites. You then create users and give them access, and they republish their workbooks and data sources. Unlike the other method, in this one, you do not reuse any of your existing Tableau Server infrastructure.

This method is most appropriate for smaller deployments with fairly autonomous and data savvy users. From an administrative perspective, this method is the simplest, since you’re not actively porting over content. However, because you rely entirely on users to republish content, this method may not be successful for large organizations or for those where centralized oversight of content is required.

**User filters**

User filters are domain-specific. Therefore, when the domain of Tableau Server changes or authentication type changes, filters no longer function as expected. Although the user filters are generated by Tableau Server, after they are set by the user, the filters are stored in the workbooks and data sources. Neither of these methods for changing the identity store modifies the contents of the workbooks or data sources.
As you plan the identity store change, you must also include a final task to correct user filtering in all workbooks and data sources with Tableau Desktop.

User names and the Tableau Identity store

If you are using Method 1, it’s helpful to understand how Tableau Server stores user names in the Tableau identity store. Tableau stores all user identities in the repository, which coordinates content permissions and site membership with various services in Tableau Server. Generally, an identity store configured for Active Directory store user names in the format, domain\username. Some organizations use a UPN (jsmith@domain.lan).

On the other hand, organizations that configure Tableau Server with local identity store usually create standard, truncated user names, such as jsmith.

In all cases, these user names are literal strings that must be unique in the Tableau identity store. If you are changing from one identity store type to another, then your target authentication, SSO, or user provisioning solution may require a specific user name format.

Therefore, to maintain all permissions, content, and user viability, one of the following must be true after you change the identity store type:

- The new user names must match the original user names, or
- The original user names must be updated to match a new format.

If an authentication change is driving the identity store change, then the target authentication scheme will likely impose a user name syntax that is different than your original user names. Method 1 includes a process where you can map original user names to new user names.

It’s possible that the original user name format will work with the new authentication type. For example, if you used UPN names in a local identity store deployment, you might be able to use the same user names in an Active Directory deployment. You could also use the domain\username format for local identity store, as long as users continue to use that format to sign in to Tableau Server.

If you are changing from local identity store to an external Active Directory store, review the topic, User Management in Active Directory Deployments, as part of your planning process.
Method 1: Use site export and import

1. Export all sites on your server. See Export or Import a Site.
2. Back up, remove, and then reinstall.
3. Create new users on Tableau Server. You should have a new user that corresponds to each user on the original server.
4. Import the sites that you exported in Step 1. See Export or Import a Site. During import, you will be prompted to map the new users to the original users.

Method 2: Fresh installation—users republish content

Even if you do not plan to port content as part of your identity store change, we recommend that you back up the server.

1. Back up, remove, and then reinstall.
2. Create users, sites, and groups.
3. Inform your users of the new Tableau Server, provide them with credentials, and allow them to republish their content.

Back up, remove, and then reinstall

Both methods include the following steps:

1. Back up Tableau Server
2. Remove Tableau Server.
3. Reinstall Tableau Server with the new identity store type.

Step 1: Back up Tableau Server

As a best practice, you should back up the server before proceeding.

Follow the procedure, Create a backup using the TSM command line interface (CLI). Run the backup command with the \(-d\) option. The \(-d\) option adds the datestamp.

When you are finished, copy the backup file (.tsbak) to a safe location that is not a part of your Tableau Server installation.
Step 2: Remove Tableau Server

You must completely remove Tableau Server from the computer. See Remove Tableau Server from Your Computer.

Step 3: Reinstall Tableau Server with new authentication type

1. Go to the Tableau Customer Portal, sign in with your Tableau user name and password, and then download Tableau Server.
2. Install Tableau Server. See Install and Configure Tableau Server more information. During installation, you will select the new identity store type. See Configure Initial Node Settings.

Migrate to New Hardware

Use the following procedure to migrate Tableau Server from one computer to another. You might do this if you are upgrading the computer hardware that Tableau is running on.

Specifically, these steps describe how to move Tableau Server data from your in-production computer to a new computer where Tableau Server is installed. Before you start, make sure you have followed the steps in Prepare for the Upgrade, including creating a backup and gathering any assets that require manual actions. For details, see Perform a Full Backup and Restore of Tableau Server. You'll need these to restore your Tableau Server data and configuration.

You can only restore from a backup that has the same type of identity store as the running server. For example, a backup from a server using local authentication can be restored to a Tableau Server initialized with local authentication, but a backup from a server using Active Directory authentication cannot be restored to a server initialized with local authentication.

**Note:** This operation includes steps that you may need to perform using the TSM command line.
1. Deactivate your product key on your original installation of Tableau Server. You need to do this so you can activate it on the new computer. For more information, see "Tableau Server product key activation" below.

2. Install Tableau Server on the new computer, activate your license, configure initial settings, and create your admin user. For details, see Install and Configure Tableau Server.

3. Copy your backup file .tsbak to the folder location specified by the base-filepath.backuprestore variable. By default this is: /var/-opt/tableau/tableau_server/data/tabsvc/files/backups.

   **Note:** You can change the location for the backup file. For more information, see tsm File Paths.

4. Next, stop Tableau Server:

   tsm stop

5. Restore your in-production data to your new Tableau Server installation using the tsm maintenance restore command:

   tsm maintenance restore -f <filename>

   where <filename> is the name of the .tsbak file you copied in step 3.

   For more information about restoring Tableau Server data, see Restore from a backup.

6. Start the server:

   tsm start

7. **Distributed installations only:** Install Tableau Server on the new computers you
want to add to your new Tableau Server cluster. See Install and Configure Additional Nodes for steps.

8. If you have not deactivated your product key on the old computer, do that after you test your new Tableau Server installation and confirm that it's ready for production. For details, see "Tableau Server product key activation" below.

Note: If you do not have an internet connection, you are prompted to create an off-line activation file to complete the deactivation process. See Activate Tableau Server Offline - Adding a License for steps.

Tableau Server product key activation

You can activate the same Tableau Server product key up to three times. This allows you to test Tableau Server (in a sandbox or QA environments, for example), as well as use Tableau in production. To maximize your activations, you should deactivate your product key when you remove Tableau Server from a computer, unless you will be reinstalling Tableau on the same computer. Doing this gives you the opportunity to use the activation on a different computer. For example, if you move Tableau Server from one computer to another, deactivate the product key, then remove Tableau from the original computer. When you install Tableau on the new computer, you can activate the key there without any conflict. If you are removing Tableau Server to reinstall it on the same computer, you don't need to deactivate the key. Tableau will use the key when reinstalled. For example, if you are moving Tableau from one drive on a computer to a different drive on the same computer. For information on how to deactivate a product key, see tsm licenses deactivate.

Migrate from Tableau Online to Tableau Server

If you want to migrate content from Tableau Online to your own self-hosted Tableau Server environment, you can request one complimentary site export from Tableau. At your request, Tableau will export a copy of your site in Tableau Online and securely deliver it to you.
You can import your site copy into your own self-hosted Tableau Server environment and recover the workbooks, projects, data sources, and the respective permissions on the content that you defined in your Tableau Online site. You cannot use a site export to keep content in-sync between Tableau Online and Tableau Server. However, you can use the Tableau Developer Tools to build your own solution for keeping your data in-sync.

Site exports require close coordination between you and Tableau’s Support and Engineering teams. It is important that you are aware of any risks or limitations to the process prior to initiating your site export. The Tableau Services offerings can help you identify these limitations, and to define a migration plan. If you need more than one export of your Tableau Online site, contact your account manager and discuss the available Tableau Services offerings. A Tableau Services architect can guide and support your custom transition to a self-hosted Tableau Server.

When can Tableau perform site exports?

Tableau can perform a site export from Tableau Online outside of major releases and weekly releases.

Major Releases

Tableau Online always runs the latest major release of Tableau Server. These releases ship on a quarterly cadence, and include new features and capabilities not found in earlier versions. This means once your source pod is upgraded, so is your site; we do not support exports of a site from Tableau Online into an earlier version of Tableau Server. For more information, see Tableau Online Upgrades.

If you plan to export your site, you must work with your Tableau Sales or Services representative to first understand the potential schedule impact from our major release activities. It is important to plan around our projected release date windows and understand which version of Tableau Server will be required on your destination site. The following diagram shows how to avoid incompatible versions during site exports.
Weekly releases

Weekly releases are shipped to Tableau Online for quickly addressing known issues. These releases are compatible with Tableau Server of the same major version. While frequent updates of Tableau Online is good for software quality, site exports cannot be performed when maintenance is scheduled for the pod on which your site resides.

Weekly releases to Tableau Online pods typically occur Thursdays, Fridays and weekends; during periods of low overall use. Because of this, our release operations generally permit site exports during the following windows:

Monday - Wednesday
10am-12pm
2pm-4pm
PST PST

Thursday – Friday*
10am-12pm
2pm-4pm
PST PST

*Feasibility of scheduling your site export on Thursdays and Fridays is dependent on your Tableau Online site’s source pod.
When you request an export from our Technical Support team, or by working with your Tableau Sales representative, our teams will coordinate a specific date and time for your export within these windows. If you require an export on a Thursday or Friday, Tableau must first verify that no releases are planned for your Tableau Online source pod on that day.

What information is (or isn’t) preserved in a site export?

The export file you receive from Tableau preserves all site content, like workbooks, projects, data sources, and users. This includes permissions set on content, user favorites lists, site quota, and other site settings. For more information, see Export or Import a Site.

Users’ custom views are preserved; however, migrating from Tableau Online to a self-managed Tableau Server will change the host name and invalidate bookmarks. After the migration is complete, you should remove existing bookmarks or update the URL path with the new Tableau Server host name.

The following Tableau Online site content is not included in a site export:

- Usage data, which appears in the site’s administrative views, is not preserved.
  - For example, view and data source activity, user actions, or performance data.
  - Usage data recorded via Admin Insights is preserved.
- OAuth access tokens embedded in data connections are reset.

  For those data sources, you will need to edit the connections and re-authenticate to the underlying data.
- Tableau Online SAML configuration settings are not preserved.
  
  SAML settings are reset and must be enabled for site-specific or server-wide SAML using Tableau Services Manager (TSM).
- Custom subscription schedules created in a Tableau Online site are not preserved on import to your self-hosted Tableau Server.

  Before you contact Tableau to request your site export, remove these custom schedules. Only the default schedules in Tableau Online are preserved.
Tableau Server on Linux Administrator Guide

- **Tableau Bridge** tasks will not be available on your self-hosted Tableau Server.

  Tableau Bridge only works with Tableau Online sites. You will need to convert Tableau Bridge connections to non-Bridge connections within your self-hosted Tableau Server.

- Date functions are localized to the time-zone of Tableau Server.

  Workbooks or data sources containing calculations based on NOW() or TODAY(), contain Relative Date Filters, or contain time stamps may show unexpected results after moving to Tableau Server if the time zone is inconsistent with the Tableau Online pod.

Prepare the Tableau Online source site and self-hosted Tableau Server site

Perform the following steps to prepare the Tableau Online source site and self-hosted Tableau Server site.

1. Set up and deploy Tableau Server (check for version compatibility).

   The most recent major version of Tableau Server must be installed on your self-hosted server prior to import. For example, if the current version of Tableau Online is 2019.2.x, your server must also have 2019.2 installed.

   You cannot import a site that is outside the target site's version family; for example, 2019.2 to 2019.1. For version information, see the [Tableau Release Notes](https://www.tableau.com/about/release-notes).

   You can find the current version in About Tableau Online under the Help menu.
After you have installed the current version, prepare the source and target sites. For more information, see Prepare the source and target sites.

2. What needs to be done in the source Tableau Online site:

   - Delete stale or unwanted content from the source site, to expedite both site export and import processes.
   - Remove obsolete or unlicensed users from the destination site to exclude them from the import, to ensure Tableau Online users properly map to Tableau Server during the verification and import steps.
   - Remove any custom subscriptions created by site users from the source site. For more information, see Enable Custom Schedules for Subscriptions.
   - Custom subscription schedules will not export and need to be re-created on the destination site post-import.
   - To remove custom subscriptions, go to Tasks, click Subscriptions, and select the check boxes for the custom subscriptions (i.e. subscriptions that do not match the Tableau Online default schedules). Then on the Actions menu, select Unsubscribe.

3. What needs to be done in the Destination Tableau Server:

   - Create the destination site in Tableau Server:
     - If more than one site exists on the instance, ensure you identify and use the correct target site ID during import (this is to ensure that the references of
published data sources are maintained).

- Create users on your self-hosted Server:
- Add and configure remaining users (after removing the obsolete/unlicensed users) to destination site on your self-hosted Server. For more information, see Add Users to a Site.
- Set up your target server for subscriptions, and add the subscriptions used on the source site. For more information, see Set Up a Site for Subscriptions.

How to request an export

Contact your Tableau Sales representative to determine if an export is the best solution for your business needs. We recommend Tableau Services to assist in the planning and execution of your site’s export-to-import path. A certified Services architect can aid in the coordination and execution of your Tableau Online site’s export, as well as ease your transition to an on-premise installation of Tableau Server.

When you are ready to proceed, Tableau Services architects will submit an export request to the Tableau Support team for scheduling. Tableau Support reviews cases based on priority level, so it may take some time before they are able to review your export request. If you want to do this independently, you can submit an export request case with Tableau Technical Support. Tableau Technical Support will process the site export request, but you’re responsible for performing all export preparation steps.

After the request is reviewed, Tableau Support will confirm if the Tableau Development team can execute an export at your preferred date and time. If not, they will supply alternative scheduling options. This coordination process may take up to 4 days to complete, so please plan accordingly.

All scheduled exports carry risk of conflicting with necessary, unscheduled maintenance to Tableau Online or Tableau Server. If your export requires rescheduling, your assigned Tableau Support representative or Tableau Services architect will contact you. For more information about Tableau Support, see Technical Support Programs.
What happens during an export?

Your users will not have access to your Tableau Online source site for the duration of the site export. The actual duration of a site export varies based on factors such as count and size of workbooks and data sources. To expedite both site export and Server import processes, delete stale or unwanted content from the source site.

You should allow for a four hour window for the export to be produced and delivered. However, users may not be locked out of their site for the entire four hours.

When the site export is complete, access to Tableau Online is restored to all licensed users. Tableau will then begin uploading a .zip file of your exported source site files to Tableau’s Secure File Transfer site. Your designated point(s) of contact will receive a password-protected link to this site, as well as the required password, which will be sent in separate emails. Secure links and corresponding passwords are valid for 7 calendar days and should not be forwarded for security reasons.

Post export steps

You will receive a .zip file containing your Tableau Online site’s data and user files. To import these files to the destination site on Tableau Server, you will need to generate .csv mapping files in Server.

Ensuring your destination site in Tableau Server is appropriately configured to receive .csv mapping files is crucial to a successful and timely site import.

Step 1: Copy the zip file into the Tableau Server directory

On your Tableau Server machine, copy the exported .zip file to the directory Tableau Server expects to find the files for importing.

Example: C:\ProgramData\Tableau\TableauServer\data\tabsvc\files\siteimports
Step 2: Generate and validate the mapping files

1. Run the following command (Tableau Server must be running):

   tsm sites import --site-id <target-siteID> --file <export-file.zip>

   This command generates a set of .csv files that show how the source site’s resources will be assigned to the target site when the import is complete.

2. Review items in the files that Tableau Server was unable to map, as indicated by a series of question marks (???), and replace with valid values from your target site.

3. Before you can complete the import process, you must replace the question marks with valid assignments on the target site.

For full information on generating and validating mapping files, see Export or Import a Site.

Step 3: Import the correctly mapped files to the target site

After you verify the site mappings in the .csv files, you can import the settings to the new site to complete the migration process.

Run the following command:

   tsm sites import-verified --import-job-dir <import-id-directory> --site-id <target-siteID>

For example:

   tsm sites import-verified --import-job-dir "C:\ProgramData\Tableau\Tableau Server-\data\tabsvc\files\siteimports\working\import_ff00_20180102022014457" --site-id new-site

When the success message appears, sign in to the new site and confirm that everything was imported as you expected.
Troubleshooting

If you continue to publish new content to your source site after the export

If you continue to publish new content to your source site after the import, you must manually republish this content or use supported tools such as Tableau’s Automation tools to facilitate moving resources from Tableau Online to Tableau Server. Tableau will not provide additional exports of Tableau Online sites to account for differences in content between source and destination sites.

If you encounter difficulty importing your site into Tableau Server

If you encounter difficulty importing your site into Tableau Server, you should file a new export request case with Tableau Technical Support, so that they can assist with troubleshooting the import process. If you have purchased Consulting Services to assist with your conversion to Server, your Tableau Services architect will also have access to your export request case.

After the exported source site files are successfully imported into your Tableau Server, the export request is considered resolved. If your Online subscription requires an extension so that users will continue to have access to your site during a protracted import to Tableau Server, contact your Sales Account Manager.

Migrate Tableau Server from Windows to Linux

Customers running Tableau Server on Windows can migrate to Linux by taking a backup of their existing Tableau installation and restoring it to a fresh installation on Linux. This topic describes the steps necessary to do this migration. You cannot migrate or upgrade from a beta version of Tableau Server to an officially released version.

The basic steps to migrate from Tableau Server on Windows to Tableau Server on Linux include:

1. Step 1: Plan your migration—Plan for your migration, including gathering all the information you'll need to be successful. During this step you should familiarize
Tableau Server on Linux Administrator Guide

Get yourself with potential differences between Tableau Server on Windows and Tableau Server on Linux.

2. Step 2: Create a backup—Create a backup of Tableau Server on Windows.

3. Step 3: Install Tableau Server on Linux and restore the Windows backup—Install a fresh instance of Tableau Server on Linux in a test environment so you can test out the migration, then restore your Windows backup. The restore of your Windows backup will restore the Tableau content (users, projects, sites, workbooks and data sources), but will not restore customizations, so you will need to spend some time configuring Tableau Server on Linux to match the expectations in your organization. If you have a multi-node installation you will need to add nodes and configure them separately.

4. Step 4: Test Tableau Server on Linux—Try Tableau Server on Linux to make sure content is there as you expect, and users are able to perform all the actions they do on Windows. Look specifically at any changes identified as potential differences between Tableau Server on Windows and Tableau Server on Linux. Include key stakeholders in the testing both to leverage their knowledge and to help communicate the upcoming changes.

5. Step 5: Install Tableau Server on Linux in your production environment and restore the Windows backup—Once you’re satisfied that Tableau Server on Linux gives you the functionality you need, install Tableau in your production Linux environment and restore the Windows backup.

Step 1: Plan your migration

A successful migration from Windows to Linux requires some preparation beforehand. You will need to satisfy the following requirements:

- **Identity store**: You can only restore from a backup that has the same type of identity store as the running server. For example, a backup from a server using local authentication can be restored to a Tableau Server initialized with local authentication, but a backup from a server using Active Directory authentication cannot be restored to a
server initialized with local authentication. To change the identity store from a local identity store to an external identity store, see Changing the Identity Store.

- **Server administrator**: You must have password for at least one of the user accounts with Server Administrator privileges from the Tableau Server source (Windows) deployment. You must use the same user as the Server Administrator in the migrated deployment to Linux.

- **Clean installation of Tableau Server for Linux**: When you install Tableau Server on Linux later in this topic, be sure to use the same identity store type as you are using on Windows, and do not create users or content. When you restore the Windows backup file to the Linux deployment of Tableau Server, all user data and content will be replaced from the Windows backup file.

- **Differences between Windows and Linux**: Review the differences between Tableau Server on Windows and on Linux so you are aware of them and can investigate in your test environment.
  
  - If you are migrating from a version of Server on Windows prior to 10.5, approach the migration like an upgrade and familiarize yourself with any changes between your existing version and 10.5 by reading What's Changed in the Server on Windows help.
  
  - Any custom fonts you use may need to be installed on your Linux computer, and may render differently there than on Windows.
  
  - Connection options for Linux are a subset of those available for Tableau Server on Windows. Review the connection types that are available and make sure the ones you need are supported.
  
  - TSM replaces tabadmin. Understand the TSM commands and how they compare to older tabadmin commands. For a list of tabadmin commands and their TSM equivalents, see Migrate from Tabadmin to the TSM CLI.
Step 2: Create a backup

The steps to create a backup depend on what version of Tableau Server on Windows you are running. If you have version 2018.1.x or earlier, you need to use tabadmin to create your backup. If you are running version 2018.2.0 or later, you use TSM to create the backup.

Migrate from 2018.1 or earlier

If you are migrating from Tableau Server on Windows version 2018.1.x or earlier:

1. Log on to the computer running Tableau Server on Windows.
2. Open a command prompt as an administrator.
3. Navigate to the bin directory. For example:

   cd "C:\Program Files\Tableau\Tableau Server\10.4\bin"

Run the following command:
4. `tabadmin backup tabserver -v -d`

   Include the `-v` flag to verify the integrity of the backup. Include the `-d` flag to include the date in the file name. In the example, `tabserver` will be used as the base file name of the resulting backup file. The date of the backup will be appended to the file name, for example, `tabserver-2017-12-20.tsbak`.

For more information, see Back Up Tableau Server Data in the Tableau Server on Windows help (version 2018.1).

Migrate from 2018.2 or later

If you are migrating from Tableau Server on Windows version 2018.2.0 or later:

1. Log on to the computer running Tableau Server on Windows.
2. Open a command prompt as an administrator.
3. Run the following command:
Tableau Server on Linux Administrator Guide

```bash
tsm maintenance backup -f <filename> -d
```

Include the `-d` flag to include the date in the backup file name.

For more information, see Back up Tableau Server data in the Tableau Server on Windows help.

**Step 3: Install Tableau Server on Linux and restore the Windows backup**

In a test environment, install Tableau Server on Linux:

- Install Tableau Server for Linux according to the procedure, Install and Configure Tableau Server. Use the same identity store as on your Windows deployment, and do not create users or content.

**Restore the Tableau Server on Windows backup:**

1. Copy the Windows backup file to the computer running Tableau Server on Linux. By default the restore process will look for the file in this location:

   ```bash
   /var/opt/tableau/tableau_server/data/tabsvc/files/backups/
   ```

   You can change the location. For more information, see tsm File Paths.

2. Run the following command to stop Tableau Server:

   ```bash
tsm stop
   ```

3. Run the following command to restore from the backup file:

   ```bash
tsm maintenance restore -f <filename.tsbak>
   ```

   Where `<filename.tsbak>` is the name of your backup file, for example, `tab-server-2017-10-20.tsbak`.

   For more information, see tsm maintenance restore

4. Run the following command to start Tableau Server:

   ```bash
tsm start
   ```
Tableau Server on Linux Administrator Guide

After restoring your Tableau content, you may need to configure Tableau Server. For example, if you are migrating from a multi-node installation, you will need to add and configure the additional nodes. You will also need to configure any customizations you made on Windows, including copying over image or logo files if applicable.

Step 4: Test Tableau Server on Linux

Once you’ve installed Tableau Server on Linux and restored your Windows backup, you can test the new version of Tableau Server. Because the restore process only restores content to Tableau, you may need to update configuration, topology (adding additional nodes for example), and any customizations you have on your Windows installation.

You’ll want to be familiar with the known differences between Windows and Linux, as well as any potential problem areas. Test basic functionality, along with any special aspects of server that your organization relies on. For example, there may be key data sources that your organization uses with Tableau. Test these to make sure you’re seeing what you expect.

These are some areas of testing to consider:

- **User access.** Confirm that Tableau Server users, including administrators, can sign in. Test your normal user sign in process. Have some of your users participate in the testing to make sure they are able to sign in as expected, and that they can get to the same content that they have access to in your production environment.

- **Viewing built-in administrative views.** With this release of Tableau Server on Linux, you must install PostgreSQL drivers manually, and the administrative views depend on these. Confirm that you have installed the drivers necessary by accessing the built-in administrative views. For more information, see Administrative Views .

- **Data source availability.** Tableau Server on Linux supports a subset of the data sources on Tableau Server on Windows. You need to confirm that the data sources used by your organization are supported on Linux, and install any drivers required. For details on which data sources are supported, see the Tableau Server tech specs. For
information about installing drivers, see Database Drivers.

- **Access to file-based data sources on shared drives.** Data sources such as Excel files on network drives will require special actions in order to be accessible from Linux. You or your IT department will need to mount the drives and update any workbooks using these data sources. For more information on connecting to network drives, see the Tableau Community.

- **Viewing published workbooks.** Have users who are familiar with the content try to view published workbooks to make sure they appear as expected. Test views embedded in web pages (for example, in SharePoint pages). Be especially aware of how fonts may differ between Windows and Linux, and some custom fonts may need to be added to your Linux computer, or replaced with other fonts if they are not available on Linux. Dashboard layouts may appear different as well, due to differences in fonts.

- **Server processes.** Sign in to Tableau Server as a server administrator, and then open the Server Status page to confirm that all services and processes are running as expected. Be aware of intentional changes due to changes in version 10.5. For more information, see Tableau Server Data Engine.

- **Publishing workbooks and data sources.** Have users publish workbooks and data sources from Tableau Desktop to make sure this goes as you expect. You may need to install drivers to support the data source connections you are using. See Database Drivers.

- **Subscriptions and extract refreshes.** Manually run some extract refreshes to confirm that they complete successfully. Run some key scheduled extract refreshes to confirm that they complete as expected.

- **Permissions.** Confirm that permissions are still set as expected for users and content.

- **Command-line utilities and APIs.** If applicable, test the command line utilities (tsm and tabcmd) and programmatic access via APIs. With Tableau Server on Linux,
tabadmin is replaced by TSM. For more information, see Migrate from Tabadmin to the TSM CLI.

Step 5: Install Tableau Server on Linux in your production environment and restore the Windows backup

When you have completed testing and have identified those areas that require additional changes on your part, or communication to your users, you are ready to install Tableau Server on Linux in your production environment and restore the Windows backup. To do this, follow the same steps described above.

**Important:** If you are migrating from Tableau Server version 2018.1 or earlier on Windows - You can take a backup from Tableau Server on Linux and restore that backup to Tableau Server on Windows version 2018.2 or later, but you cannot restore that backup to Tableau Server on Windows version 2018.1 or earlier. This means that if you migrate to Linux, you cannot easily switch back to Tableau Server on Windows version 2018.1 or earlier. Keep this in mind as you test your migration, and consider keeping your installation on Windows until you are certain that you no longer need it.

Test Workbook Performance

In most cases, you can expect to see similar or better performance after upgrading your Tableau Server from versions 10.4 or earlier to 10.5 or later. However, for workbooks that use complex calculations, you may see performance issues after you upgrade. This article describes how to test performance and troubleshoot these performance issues before you upgrade your production Tableau Server from versions 10.4 and earlier to 10.5 or later.

If you have already upgraded your production Tableau Server to 10.5 or later, and you are seeing performance issues for workbooks that have calculations and use extracts as the data source, see Troubleshoot Performance of Workbooks with Calculations.

Since Linux was first released in 10.5, this only applies when you are migrating from Tableau Server versions 10.4 or earlier on Windows to Linux 10.5 or later.
To make sure your workbook performance is same or better after your upgrade to Tableau 10.5 or later, we strongly recommend that you set up a test environment and do performance testing before you upgrade your production Tableau Server.

Use the following sequence of steps for testing workbook performance:

1. **Set up your test environment**

2. **Capture performance metrics and analyze**

3. **Troubleshoot performance issues**

4. **Capture the conclusions and results**

Set up your test environment

1. Create a test environment that mirrors your production environment as closely as possible.

   Follow the steps described in Migrate Tableau Server from Windows to Linux.

2. Disable any automated or scheduled extract refresh schedules.

Capture performance metrics and analyze

1. Run Tabjolt or any other tool of your choice to capture performance metrics on your current version of Tableau Server to set a baseline. Use workbooks that have calculations and use extracts as the data source for testing. For more information about how install and use Tabjolt, see the Tabjolt Installation Guide.

2. Perform a full backup to capture the metrics from the first step.

3. Upgrade your test environment to 10.5.3 or later. We are recommending 10.5.3 as the minimum version since the settings you need to enable in order to resolve performance issues later on are only available in 10.5.3 or later versions of Tableau.

4. **Checkpoint:** Run Tabjolt to compare workbook performance before and after upgrade.
Tableau Server on Linux Administrator Guide

If you see performance degradation after upgrade at this step, it may be due to other Tableau Server issues and might require further investigations. For more information, see Record and Analyze Workbook Performance.

If you see no difference in performance before and after upgrade, continue to the next step.

5. Select a subset of workbooks and run a full refresh of the extracts. Doing a full refresh will upgrade the extracts from .tde to .hyper.

6. **Checkpoint:** Run the tool again to compare the performance of your workbooks between step 4 and step 6 which is essentially before and after doing extract refreshes. Specifically, check for workbooks with slower response times or failure to load.

Troubleshoot performance issues

If you find that you are unable to test due to performance issues, or you find that VizQL server is consuming all or most of the available memory on the machine while viewing them, follow **steps 1 through 3**.

For workbooks that load successfully, but have slower response after upgrade, follow **steps 4 and 5**.

1. Enable the following setting that checks the number of query nodes required for Tableau to generate the workbook and displays an error when the set limit is exceeded. This helps prevent VizQL process from consuming the memory resources on the machine that might result in critical Server issues:

   **Use TSM CLI to run the following commands:**

   ```
   tsm configuration set -k native_api.node_limit_checker_pre_rewrite_disable -v false
   tsm pending-changes apply
   ```
Note: This setting is only available in Tableau Server 10.5.3 or later. While enabling this setting improves the reliability and stability of Tableau Server, you may see errors on a very small percentage of workbooks with when you enable this setting. For example, the Tableau team saw errors on 0.01% of the workbooks with this setting enabled on Tableau Online.

2. Test the workbooks by viewing or interacting with them. You should see an error message informing you about exceeding the node limits on workbooks that have performance issues - The error message might say something like: "Logical Query tree has 2348182 nodes, maximum number allowed is 1000000...."

3. Use Tableau Desktop to troubleshoot your workbook performance. Run this command to enable the node limit check in your Tableau Desktop:

   `tableau.exe -DNodeLimitCheckerPreRewriteDisable=false`

   **Note:** This setting is only available in Tableau Desktop 10.5.3 or later.

   In Tableau Desktop, download and open the workbooks that displayed the error message about high node limits in step 2. Try optimizing the performance of workbooks using methods described in the Best Practices for Creating Calculations in Tableau topic. If successful, then publish them to Tableau Server. If you are still experiencing performance issues, continue to the next step.

4. For workbooks that load successfully, but the overall response times is slower after upgrade, materialize calculations for your extract:

   Using Tabcmd:

   `tabcmd refreshextracts --workbook "My Workbook" --addcalculations`

   For more information, Compute Calculations Now option.
5. If you have a large number of workbooks that fall under the above category (overall response times is slower after upgrade), consider enabling the server wide setting to retain materialized calculations for all extracts when they are upgraded from .tde to .hyper:

**Using TSM CLI:**

```bash
 tsm configuration set -k native_api.preserve_calculations_on_hyper_refresh_conversion -v true
 tsm pending-changes apply
```

**Considerations for turning on this setting:**

Turn this setting on only if majority of your workbooks require calculations to be materialized. There are resource and time costs associated with this setting, as described below:

- When this setting is set to true, the extract file size will increase, affecting the overall disk space.

- Extracts with materialized calculations take a longer time to refresh than extracts without materialized calculations.

This setting only affects extracts that are not yet upgraded to .hyper. Extracts already upgraded to .hyper before turning on this setting will not have the calculations materialized. This setting will also not affect workbooks and extracts published from Tableau Desktop, or any new extract refreshes published to Tableau Server that do not use the **Compute Calculations Now** option.

**Note:** This setting is only available in Tableau Server 10.5.3 or later.

6. Enable the extract refresh schedules and monitor the performance of your workbooks to make sure that your workbooks are performing as expected.
Capture the conclusions and results from your testing

Make a note of the settings and steps that helped resolve the performance issues to use them to prevent performance degradation when you upgrade your Production Tableau Server.

For example, if during your testing, you found that you needed to set the `native_api.preserve_calculations_on_hyper_refresh_conversion` setting to true, remember to disable your extract refreshes just before you upgrade your production Tableau Server.

After upgrade, set the `native_api.preserve_calculations_on_hyper_refresh_conversion` setting to true, and then re-enable your extract refreshes after you have confirmed that you are not experiencing slow response times after upgrade.

Migrate from Tabadmin to the TSM CLI

The Tableau Services Manager (TSM) command-line interface (CLI) replaces the tabadmin CLI in Tableau Server on Linux, and in Tableau Server on Windows version 2018.2. This page maps tabadmin commands to TSM commands to help you to migrate to the TSM CLI.

To learn more about the TSM CLI, see tsm Command Line Reference.


Tabadmin commands with a corresponding TSM CLI command

The following table shows which tabadmin commands correspond to commands available in the TSM CLI.

<table>
<thead>
<tr>
<th>Command Description</th>
<th>Tabadmin Command(s)</th>
<th>Comparable TSM CLI Command</th>
</tr>
</thead>
</table>

Version: 2019.4
### Tableau Server on Linux Administrator Guide

<table>
<thead>
<tr>
<th>Task</th>
<th>Command 1</th>
<th>Command 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Activate a license</td>
<td><code>tabadmin activate</code></td>
<td><code>tsm licenses activate</code></td>
</tr>
<tr>
<td>Deactivate licenses</td>
<td><code>tabadmin activate -- return</code></td>
<td><code>tsm licenses deactivate</code></td>
</tr>
<tr>
<td>Activate a trial license</td>
<td><code>tabadmin activate -- trial</code></td>
<td><code>tsm licenses activate -trial</code></td>
</tr>
<tr>
<td>Create a backup of the data managed by Tableau Server</td>
<td><code>tabadmin backup</code></td>
<td><code>tsm maintenance backup</code></td>
</tr>
<tr>
<td><strong>Note:</strong> A backup created using TSM does not include any server configuration data. There is no option to include server configuration data.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Clear the server cache</td>
<td><code>tabadmin clearcache</code></td>
<td><code>tsm maintenance cleanup -r</code></td>
</tr>
<tr>
<td>Clean up temporary files and old log files</td>
<td><code>tabadmin cleanup</code></td>
<td><code>tsm maintenance cleanup</code></td>
</tr>
<tr>
<td>Update the server configuration with any changes you’ve made</td>
<td><code>tabadmin configure</code></td>
<td><code>tsm pending-changes apply</code></td>
</tr>
<tr>
<td>Customize the server name and logos</td>
<td><code>tabadmin customize</code></td>
<td><code>tsm customize</code></td>
</tr>
<tr>
<td>Enable access to the repository</td>
<td><code>tabadmin dbpass</code></td>
<td><code>tsm data-access repository-access enable</code></td>
</tr>
<tr>
<td>Operation</td>
<td>Command 1</td>
<td>Command 2</td>
</tr>
<tr>
<td>---------------------------------------------------------------------------</td>
<td>------------------------------------------------</td>
<td>------------------------------------------------</td>
</tr>
<tr>
<td>Disable access to the repository</td>
<td><code>tabadmin dbpass --disable</code></td>
<td><code>tsm data-access repository-access disable</code></td>
</tr>
<tr>
<td>Set a file store instance to read-only mode</td>
<td><code>tabadmin decommission</code></td>
<td><code>tsm topology filestore decommission</code></td>
</tr>
<tr>
<td>Delete one or more Web Data Connectors (WDCs) from Tableau Server</td>
<td><code>tabadmin delete_web-dataconnector</code></td>
<td><code>tsm data-access web-data-connectors delete</code></td>
</tr>
<tr>
<td>Add a Web Data Connector (WDC) to Tableau Server</td>
<td><code>tabadmin import_web-dataconnector</code> and</td>
<td><code>tsm data-access web-data-connectors add</code></td>
</tr>
<tr>
<td></td>
<td><code>tabadmin whitelist_web-dataconnector</code></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Note:</strong> TSM does not support importing WDCs, instead it lets you add WDCs to a safe list (or &quot;whitelist&quot;). To learn more, see Web Data Connectors in Tableau Server.</td>
</tr>
<tr>
<td>List Web Data Connectors (WDCs) used by Tableau Server</td>
<td><code>tabadmin list_web-dataconnectors</code></td>
<td><code>tsm data-access web-data-connectors list</code></td>
</tr>
<tr>
<td></td>
<td></td>
<td>To learn more, see Web Data Connectors in Tableau Server.</td>
</tr>
<tr>
<td>Export a site from Tableau Server</td>
<td><code>tabadmin exportsite</code></td>
<td><code>tsm sites export</code></td>
</tr>
<tr>
<td>Initiate a repository failover</td>
<td><code>tabadmin fail-</code></td>
<td><code>tsm topology failover-</code></td>
</tr>
<tr>
<td>Command</td>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>--------------------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td></td>
</tr>
<tr>
<td><code>tabadmin get</code></td>
<td>Get a configuration option</td>
<td></td>
</tr>
<tr>
<td><code>tsm configuration get</code></td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>tabadmin get_openid_redirect_url</code></td>
<td>Get the OpenID redirect URL</td>
<td></td>
</tr>
<tr>
<td><code>tsm authentication openid_get-redirect-url</code></td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>tabadmin importsite</code></td>
<td>Import site .csv files into Tableau Server</td>
<td></td>
</tr>
<tr>
<td><code>tsm sites import</code></td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>tabadmin importsite_verified</code></td>
<td>Import a site into Tableau Server using .csv files</td>
<td></td>
</tr>
<tr>
<td><code>tsm sites import-verified</code></td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>tabadmin licenses</code></td>
<td>Display license information for Tableau Server</td>
<td></td>
</tr>
<tr>
<td><code>tsm licenses list</code></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Note:</strong> For more information about the output of this command, see View Server Licenses.</td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>tabadmin recommission</code></td>
<td>Move a file store from read-only mode to an active read/write state</td>
<td></td>
</tr>
<tr>
<td><code>tsm topology filestore recommission</code></td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>tabadmin regenerate_internal_tokens</code></td>
<td>Regenerate internal security tokens</td>
<td></td>
</tr>
<tr>
<td><code>tsm security regenerate-internal-tokens</code></td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>tabadmin register</code></td>
<td>Register Tableau Server</td>
<td></td>
</tr>
<tr>
<td><code>tsm register</code></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Task</td>
<td>Command 1</td>
<td>Command 2</td>
</tr>
<tr>
<td>-------------------------------------------</td>
<td>-------------------</td>
<td>-------------------</td>
</tr>
<tr>
<td>Rebuild the search index for Tableau Server</td>
<td><code>tabadmin reindex</code></td>
<td><code>tsm maintenance rein-index-search</code></td>
</tr>
<tr>
<td>Reset the Tableau Server administrator account</td>
<td><code>tabadmin reset</code></td>
<td><code>tsm reset</code></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Note:</strong> Added in version 2018.1.</td>
</tr>
<tr>
<td>Stop and restart all Tableau Server processes</td>
<td><code>tabadmin restart</code></td>
<td><code>tsm restart</code></td>
</tr>
<tr>
<td>Restore from a Tableau Server backup file</td>
<td><code>tabadmin restore</code></td>
<td><code>tsm maintenance restore</code></td>
</tr>
<tr>
<td></td>
<td></td>
<td>The restore command does not restore any server configuration data. This is true whether you are using a backup created with TSM or a backup created with <code>tabadmin</code>.</td>
</tr>
<tr>
<td>Set a configuration option</td>
<td><code>tabadmin set</code></td>
<td><code>tsm configuration set</code></td>
</tr>
<tr>
<td>Activate or suspend a site</td>
<td><code>tabadmin sitestate</code></td>
<td><code>tsm sites unlock</code></td>
</tr>
<tr>
<td>Start all Tableau Server processes</td>
<td><code>tabadmin start</code></td>
<td><code>tsm start</code></td>
</tr>
<tr>
<td>Get the status of Tableau</td>
<td><code>tabadmin status</code></td>
<td><code>tsm status</code></td>
</tr>
</tbody>
</table>
Stop all Tableau Server processes

Create an archive (.zip) file with Tableau Server log files

Tabadmin commands with no corresponding TSM CLI command

<table>
<thead>
<tr>
<th>Command Description</th>
<th>Tabadmin Command</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add or remove a user from the system administrator group</td>
<td><code>tabadmin administrator</code></td>
<td>You can use the Tableau Server REST API Add User to Group and Remove User from Group methods to add or remove a user from the system administrator group.</td>
</tr>
<tr>
<td>Create a new key to encrypt sensitive information stored in the repository</td>
<td><code>tabadmin assetkeys</code></td>
<td>Use the <code>tsm security regenerate-internal-tokens</code> command to create or regenerate secrets and master keys.</td>
</tr>
<tr>
<td>Specify whether Tableau Server starts at system</td>
<td><code>tabadmin autostart</code></td>
<td>Tableau Server returns to the state it was in prior to a system restart. If it was running, it will</td>
</tr>
<tr>
<td>start-up time</td>
<td>restart. If it was stopped it will be stopped after the system starts.</td>
<td></td>
</tr>
<tr>
<td>Identify a second server node for backup</td>
<td><code>tabadmin fail-overprimary</code></td>
<td>TSM does not have primary nodes, so a TSM equivalent to this command is not needed.</td>
</tr>
<tr>
<td>Manage credentials for delegated data access on Tableau Server</td>
<td><code>tabadmin manage_global_credentials</code></td>
<td>We recommend that you use Kerberos delegation to Apache Impala for global credential management. To learn more, see Kerberos and Enable Kerberos Delegation for Hive/Impala in the Tableau Community.</td>
</tr>
<tr>
<td>Reset the password for a Tableau Server account</td>
<td><code>tabadmin passwd</code></td>
<td>If your server uses local authentication, you can use the Tableau Server REST API Update User method to reset the password for a user account.</td>
</tr>
<tr>
<td>Reset binding between Tableau Server user ID and Open ID Connect identity provider</td>
<td><code>tabadmin reset_openid_sub</code></td>
<td></td>
</tr>
<tr>
<td>Determine whether your environment meets the minimum requirements to run Tableau Server</td>
<td><code>tabadmin validate</code></td>
<td></td>
</tr>
</tbody>
</table>
## Verify that a backup of the Tableau Server repository will restore successfully

**Tabadmin verify_database**

**The tsm maintenance backup command automatically verifies that a backup will restore correctly unless you use the --skip-verification parameter.**

## Prepare VizQL processes for fast load times after a Tableau Server restart

**Tabadmin warmup**

**The tabadmin warmup command is no longer necessary, as Tableau Server is now optimized to automatically provide fast load times after a server restart.**

### Migrate Tableau Server from an On-Premises Computer to a VM in the Cloud

You can migrate Tableau Server from a computer in your data center to a virtual machine (VM) in the cloud. As a part of this migration, you’ll need to move various Tableau Server data and configuration settings from your on-premises computer to a VM in the cloud where Tableau Server is installed.

**Note:** This operation includes steps that you may need to perform using the TSM command line.

### Prerequisites

Before you migrate Tableau Server from an on-premises computer to a VM in the cloud, be sure to review the following prerequisites:
Tableau Server on Linux Administrator Guide

- Have an account and an environment set up on your cloud provider. For more information, see:
  - Install Tableau Server on Amazon Web Services
  - Install Tableau Server on the Google Cloud Platform
  - Install Tableau Server on Microsoft Azure
  - Install Tableau Server in the Alibaba Cloud
- Read Prepare for the Upgrade and Back up Tableau Server data to prepare for the migration.

To migrate Tableau Server to a VM in the cloud

1. Ensure that there are no background tasks running, such as, extract refreshes, workbook publishing, etc. When Tableau Server is idle, note any data source connection types in use so that you can install the appropriate drivers on the new Tableau Server on the VM in the cloud.
2. Stop Tableau Server. At a command prompt, type:

   tsm stop

3. After Tableau Server is stopped, make a complete backup, following the steps in Prepare for the Upgrade and Back up Tableau Server data. The backup is what you will use to restore your Tableau Server data. You’ll use this backup to restore your Tableau Server data on the VM in the cloud.
4. Move the backup off the server and store the backup securely. For example, you can store your backup on cloud storage such as Amazon S3, Google Drive, etc.
5. If you are using SSL on the on-premises server, make a copy of the SSL folder. Be sure to note any security rules, for example, firewall rules, ports, etc. to avoid communication issues between different elements of your infrastructure and your new Tableau Server on the VM in the cloud.
6. Remove any DNS entries using the IP address of the on-premises Tableau Server, so you can reassign them to the Tableau Server on the VM in the cloud.
7. Launch a VM into the cloud that meets the Tableau Server recommendations.
8. Install Tableau Server on the VM in the cloud. For more information, see:
   - Install Tableau Server on Amazon Web Services
   - Install Tableau Server on the Google Cloud Platform
   - Install Tableau Server on Microsoft Azure
   - Install Tableau Server in the Alibaba Cloud

9. After installation is complete, copy any SSL files to Tableau Server on the VM in the cloud, and restore the backup on your new server.

10. Configure Tableau Server on the VM in the cloud and install all SSH and port related stuff.

   **Note:** If you change the IP address/port of the database, you might need to update all connection information.

11. Copy your backup file `.tsbak` to the folder location specified by the `base filepath.backuprestore` variable. By default this is: `/var/-opt/tableau/tableau_server/data/tabsvc/files/backups`.

   **Note:** You can change the location for the backup file. For more information, see `tsm File Paths`.

12. Next, stop Tableau Server. At a command prompt, type:

    `tsm stop`

13. Restore your in-production data without configuration information to your new Tableau Server installation. At a command prompt, type:

    `tsm maintenance restore -f <filename>`

    where `<filename>` is the name of the `.tsbak` file you copied in step 11.
For more information about restoring Tableau Server data, see Restore from a backup.

14. Start the server. At a command prompt, type:

   tsm start

15. **Distributed installations only:** Install Tableau Server on the new VMs you want to add to your new Tableau Server cluster in the cloud. For more information, see:

   - Install and Configure Additional Nodes
   - Self-Deploy Tableau Server on AWS in a Distributed Environment
   - Self-Deploy Tableau Server on the Google Cloud Platform in a Distributed Environment
   - Self-Deploy Tableau Server on Microsoft Azure in a Distributed Environment
   - Self-Deploy Tableau Server on Alibaba Cloud in a Distributed Environment

16. Test your Tableau Server on your VM in the cloud to ensure that it works as expected. If your Tableau Server is working fine in the cloud, you can deactivate your on-premises Tableau Server product key and use this same product key to activate your Tableau Server on the VM in the cloud.

   **Note:** If you do not have an internet connection, you are prompted to create an offline activation file to complete the deactivation process. For more information, see Deactivate Tableau Server Offline.

17. The same Tableau Server product key can be activated three times: once for a production environment, once for a test environment, and once for a QA environment. After you have tested your new Tableau Server installation and confirmed that it's ready for production, you must deactivate your earlier production version of Tableau Server, and then you must uninstall it. To deactivate the earlier version, see tsm licenses deactivate.
Upgrade Tableau Server Overview

The articles in this section help you upgrade an existing installation of Tableau Server. They describe the recommended steps of planning and testing before actually performing the upgrade. There's information about best practices, and when you're ready to actually perform your upgrade, steps for upgrading a single node server and a multi-node installation. Where possible, we call out possible pitfalls and help you to avoid these.


Important: Your upgrade steps depend on which version you are upgrading from. When you are ready to actually upgrade, be sure you follow the procedure that applies to your installation:

- Upgrading from Tableau Server on Linux version 10.5 (10.5.0 or later) requires special steps. If you are upgrading from version 10.5, use the Upgrade Tableau Server on Linux from 10.5 procedure.

- Upgrading from Tableau Server on Linux version 2018.1 or later is more straightforward. If you are upgrading from version 2018.1.0 or later, follow the steps in Upgrade Tableau Server on Linux.

For instructions on how to determine your version of Tableau Server, see View Server Version.
Upgrade Tableau Server on Linux

The articles in this section help you upgrade and existing installation of Tableau Server. They describe the recommended steps of planning and testing before actually performing the upgrade. There's information about best practices, and when you’re ready to actually perform your upgrade, steps for upgrading a single node server and a multi-node installation. Where possible, we call out possible pitfalls and help you to avoid these.

**Important**: Your upgrade steps depend on which version you are upgrading from. When you are ready to actually upgrade, be sure you follow the procedure that applies to your installation:

- Upgrading from Tableau Server on Linux version 10.5 (10.5.0 or later) requires special steps. If you are upgrading from version 10.5, use the Upgrade Tableau Server on Linux from 10.5 procedure.

- Upgrading from Tableau Server on Linux version 2018.1 or later is more straightforward. If you are upgrading from version 2018.1.0 or later, follow the steps in Upgrade Tableau Server on Linux.

For instructions on how to determine your version of Tableau Server, see View Server Version.


Research the Upgrade

Before you upgrade Tableau Server, we recommend that you plan the upgrade.
Learn about the new version of Tableau Server, including what's new and what's changed.

- Search for "What's New" in the Tableau Server on Linux Help.

- Make sure the computers you are going to upgrade (both for the test environment and the production environment) meet the minimum hardware requirements. Minimum requirements and recommendations can change from version to version.

- Minimum Hardware Requirements and Recommendations for Tableau Server

  You may need to allocate additional disk space depending on the various factors like whether or not you will be using extracts. For more information, see Disk Space Requirements.

**Important:** As a best practice, you should never install a beta version of Tableau Server in your production environment. You should also never restore a production Tableau Server installation using a backup of a beta version.

- Confirm licensing requirements

  The computers you are going to upgrade (both for the test environment and the production environment) must meet these licensing requirements:

  - Trial licenses—When upgrading from one major version to another (version 10.x to version 2018.x for example), you must have a valid license. You cannot directly upgrade to a new major version if your server is running a trial license. If you have a trial license and want to upgrade to a new major version, follow these steps:

    1. Backup of your existing Tableau Server. For details, see Back up Tableau Server data.

    2. Install the new Tableau Server version on a separate computer and
activate a trial license.

3. Restore your backup to the new version. For more information, see Restore from a backup.

- Expired maintenance—You should not upgrade your server if your maintenance is expired. If you attempt to upgrade a server with expired maintenance the upgrade may fail, or, if it succeeds, may result in an unlicensed server. You can refresh your maintenance before upgrading:
  - If you are upgrading Tableau Server on Windows version 2018.1 or earlier, use the Manage Product Keys application. For more details, see Refresh Maintenance Date for the Product Key.
  - If you are upgrading Tableau Server on Windows version 2018.2 or later, or Tableau Server on Linux, run this tsm command:

    ```
    tsm licenses refresh
    ```

    If refreshing the product key does not update maintenance with a valid date, check in the Tableau Customer Portal to confirm you have a Tableau Server product key with up-to-date maintenance.

- Understand how version compatibility might impact your installation of Tableau Server.
  - Desktop and Server Compatibility in the Tableau Help.

- Add-on licenses—Data Management Add-on license is required if you want to use Tableau Prep Conductor. Tableau Prep Conductor enables publishing, running and scheduling flows on Tableau Server. For more information, see Licensing Tableau Prep Conductor.
What's New in Tableau Server

This article lists what's new in Tableau Server, beginning with version 10.5. Version 10.5 was the first version of Tableau Server for Linux.

What's new in version 2019.4

User

- Request access

New features and enhancements related to web authoring and interacting with views on the web are listed in What's New in Tableau for Users and Web Authoring and Tableau Desktop Feature Comparison. For topics related to web authoring and interacting with views, see Using Tableau on the Web.

Server or site administrator

- Configure recommendations for views
- Enable or disable Ask Data for a site
- Personal access tokens
- Secure SMTP
- Test Email Configuration
- Create Extracts on the Web
- New Installers for Tableau Server and tabcmd
- Custom welcome banner
- Limit visibility of user information
- Manage Sandboxed Extensions

Request access

Users can now request access to see a project, workbook, or view when they don’t have viewing permission. When someone requests access to content, Tableau routes an email to the person who controls permissions for that content with information about the request and
instructions on making access changes. For information on managing permissions, see Permissions.

Configure recommendations for views

Tableau Server now recommends views to help users discover relevant content. Recommendations for views are an expansion of the system that provides recommendations for data sources and tables.

- To control whether the recommendations feature is enabled for your server, use the `tsm configuration set options recommendations.enabled` and `recommendations.vizrecs.enabled`. For more information, see `tsm configuration set Options`.
- To schedule when the recommendations trainer runs on your server, adjust the server settings under `Recommendations Training Schedule`. For more information, see `Server Settings (General)`.
- To hide all recommendations on a particular site, or to hide user names from appearing on the recommendation tooltips for a particular site, adjust the site settings under `Recommendations for Views`.

Enable or disable Ask Data for a site

As a site administrator, you can control the availability of Ask Data in the `General` area of site settings. For more information, see Enable Ask Data for Sites and Data Sources.

Personal access tokens

Personal access tokens provide Tableau Server users the ability to create long-lived authentication tokens. The tokens allow users to run automation with Tableau REST APIs without requiring hard-coded credentials or interactive login. More information about using personal access tokens with Tableau REST APIs is at `Signing In and Out (Authentication)`.

As an administrator, you can revoke and track usage of the tokens. See Personal Access Tokens.
Secure SMTP

You can now configure Tableau Server to connect with your email server over a secure connection. You can encrypt and optionally authenticate with a certificate that you upload and configure on Tableau Server. See Secure SMTP.

Test Email Configuration

After you configure an SMTP connection, use the new command, `tsm email`, to validate it.

Create Extracts on the Web

Now you can create extracts in the browser, without using Tableau Desktop. For more information, see Create Extracts on the Web. You can also use the new `createextracts` and `deleteextracts` commands in `tabcmd` to create or delete extracts for a published workbook or data source. For more information, see `tabcmd` Commands.

New Installers for Tableau Server and `tabcmd`

Tableau Server 2019.4 introduces completely rewritten Setup programs for both Tableau Server and `tabcmd`. For more information, see Version 2019.4 includes the following changes you should know about before upgrading.

Custom welcome banner

You can add a custom message to the welcome banner on the home page for all server users to see. Add up to 240 characters of text and hyperlinks to provide important support information or resource to help server users get started. You can also turn off the default Tableau home banner for all server users. For more information, see Customize Your Server.
Limit visibility of user information

The User Visibility site setting lets administrators manage if users with Viewer and Explorer site roles see other users and groups on the site, which can be important for sites that are used by multiple clients. When User Visibility is set to **Limited**, Explorers and Viewers cannot see information about other users, including aliases, project ownership, comments, and more. Creators and administrators (including Site Administrator Explorers) can still view user information. By default, User Visibility is set to Full. For more information, see Manage Site User Visibility.

Manage Sandboxed Extensions

Tableau now supports Sandboxed extensions, which are a new type of dashboard extension that run in a protected environment hosted by Tableau. Like Network-enabled extensions, Sandboxed extensions are web applications that run in custom dashboard zones and can interact with the rest of the dashboard using the Tableau Extensions API. While Network-enabled extensions have full access to the web, Sandboxed extensions can’t share data or make network calls outside of the hosting Tableau Server. Sandboxed extensions provide a new level of security for Tableau users and administrators.

To learn more about controlling extensions, see Manage Dashboard Extensions in Tableau Server. To learn more about extension security and strategies for deployment, see Extension Security - Best Practices for Deployment.

Added in previous versions

Added in version 2019.3

User

- Add PDF Attachments to Subscription Emails
- Tableau Catalog features—part of the Data Management Add-on
New features and enhancements related to web authoring and interacting with views on the web are listed in What’s New in Tableau for Users and Web Authoring and Tableau Desktop Feature Comparison. For topics related to web authoring and interacting with views, see Using Tableau on the Web.

**Server or site administrator**

- Support for Embedded Credentials in Bootstrap Files
- Tableau Catalog on the server—part of the Data Management Add-on
- Tableau Metadata API
- Server Management Add-on
- Support for Debian 9 Linux Distributions
- Extract Encryption at Rest
- Date-range Support for Ziplogs
- Tableau Server Usage Data
- Add a Note to Tableau Server Sign In Page
- Support added for Italian
- Job Management

**Add PDF Attachments to Subscription Emails**

You can give your users the option to attach PDF renderings for views and workbooks subscription emails. You must enable attachment functionality on the server. See Configure Server Event Notification.

**Tableau Catalog features—part of the Data Management Add-on**

Tableau Catalog is included in the separately licensed Data Management Add-on for Tableau Server and Tableau Online, making a variety of features available to you in the data management space. You can use Tableau Catalog to discover data, curate data assets, communicate data quality, perform impact analysis, and trace the lineage of data used in Tableau content. For more information, see About Tableau Catalog.
Tableau Catalog on the server—part of the Data Management Add-on

Tableau Catalog is included in the separately licensed Data Management Add-on. For more information, see License the Data Management Add-on.

You can give your users access to metadata and features like expanded search, data details, and lineage when you Enable Tableau Catalog. To learn more about how Tableau Catalog and metadata permissions work, see Manage Permissions for External Assets.

Tableau Metadata API

If you’re a developer and interested in accessing metadata indexed by or surfaced through Tableau Catalog, see the Tableau Metadata API Help.

Job Management

You can now view, monitor, and manage backgrounder jobs on Tableau Server. For more information, see Managing Jobs in Tableau Server.

Support for Embedded Credentials in Bootstrap Files

You can now include embedded credentials when generating bootstrap file. This allows you to add a new node to your Tableau Server installation from the command line without needing to provide TSM administrative credentials when installing the node, and can be especially useful if you automate installation of Tableau. This feature is enabled by default and the embedded credentials are valid for a predetermined amount of time (see tabadmincontroller.auth.expiration.minutes) and serve as a session cookie.

Embedded credentials are designed to provide you with flexibility, so you can choose how you use this feature. You can choose to not include embedded credentials any time you generate a bootstrap file. You can also disable the feature entirely using the
features.PasswordlessBootstrapInit option with the tsm configuration set command. When credentials are not included, you are prompted, when adding a node, for the name and password of a user with administrative permissions on the computer where Tableau Services Manager is installed.

**Important:** The bootstrap file contains a copy of the master keystore file used for encrypting the configuration secrets. The file can also embedded credentials which are valid for a predetermined amount of time (see tabadmincontroller.auth.expiration.minutes) and serve as a session cookie. We strongly recommend that you take additional measures to secure the bootstrap file using mechanisms as described in Securing secrets for import and export operations.

For more information, see tsm topology nodes get-bootstrap-file.

**Server Management Add-on**

Server Management Add-on is a separately licensed Add-on which will provide enhanced security, manageability, and scalability capabilities for Tableau Server.

- **Improved scalability:**
  - For Tableau Server deployments hosted on AWS, you can now choose to host your metadata repository externally with Amazon RDS Postgres. Amazon RDS offers scalability, reliability, high availability and security built-in for PostgreSQL. By integrating with AWS, you will be able to take advantage of these additional benefits of deploying the cloud. For more information, see Tableau Server External Repository.
  
  - You can also tune your Tableau Server processes by customizing which nodes process data engine queries and background jobs. For more information, see Workload Management through Node Roles.

- **Increased operational efficiency:**
• A new Content Migration Tool which was acquired from InterWorks is included as part of Server Management Add-on.

The Content Migration Tool helps streamline your workbook and data source content work flows. You can manage the movement of content between development and production or between other environments. For more information, see About Tableau Content Migration Tool

• Enhanced security:

If your organization will be enabling extract encryption at rest, you can also integrate Tableau Server with AWS Key Management Service to provide an enhanced level of integration and security. For more information, see Key Management System.

Support for Debian 9 Linux Distributions

Beginning with Tableau Server 2019.3, Debian 9 (and later) Linux distributions are supported. See Before you install....

Extract Encryption at Rest

Extract encryption at rest is a data security feature that allows you to encrypt .hyper extracts while they are stored on Tableau Server.

Tableau Server administrators can enforce encryption of all extracts on their site or allow users to specify to encrypt all extracts associated with particular published workbooks or data sources. See Extract Encryption at Rest.

Date-range Support for Ziplogs

Two new date options for the tsm maintenance ziplogs command, --startdate and --enddate, give you the ability to specify a date range for the logs being gathered. These options must be used together, and cannot be used with the --minimumdate option. For more information, see tsm maintenance ziplogs.
Tableau Server Usage Data

Beginning with Tableau Server 2019.3, server usage data is sent to Tableau by default. This is the case for new installs and upgrades and can be disabled. You can opt out during installation, upgrade, or at any time after Tableau Server has been installed. For more information about usage data, see Usage data is sent to Tableau in the What’s Change article, and the general overview, Server Usage Data.

Add a Note to Tableau Server Sign In Page

You can add an optional note and URL to your server’s sign in page to help new users gain server access or for server-wide announcements. The setting also supports emojis. To set a custom note, sign in to a site on Tableau Server. On the left-side navigation pane, select Manage all sites from the drop-down site list. Select Settings to and add a message to Sign In Customization. For more information about customizing Tableau Server, see Customize Your Server.
Support added for Italian

Tableau has added support for Italian. You can now set Tableau to display the user interface in one of eleven languages:

- Chinese (Simplified)
- Chinese (Traditional)
- English (United Kingdom)
- English (United States)
- French
- German
- Italian
- Japanese
- Korean
- Portuguese
- Spanish

For more information, see Language and Locale for Tableau Server.

In addition to supporting Italian in all Tableau products, the help is also available in these languages. To read the help in your preferred language, navigate to the bottom of the main Tableau Help page, and select the language from the footer.

Added in version 2019.2
Tableau Server on Linux Administrator Guide

A blue Send feedback icon is displayed in the bottom right corner of every Help page to make it easier for you to send feedback about the content in a topic. Click the icon to let us know if a topic is helpful to you, and also to comment on what needs improvement.

What is a site

You might be used to using the term site to mean “a collection of connected computers,” or perhaps as the short form of “website.” In Tableau-speak, we use site to mean a collection of users, groups, and content (workbooks, data sources) that’s walled off from any other groups and content on the same instance of Tableau Server. Another way to say this is that Tableau Server supports multi-tenancy by allowing server administrators to create sites on the server for multiple sets of users and content.

User

- New features and enhancements related to web authoring and interacting with views on the web are listed in What's New in Tableau for Users and Web Authoring and Tableau Desktop Feature Comparison. For topics related to web authoring and interacting with views, see Using Tableau on the Web.

Server or site administrator

- Direct access with TSM CLI
- Local authentication password policy improvements
- Select from ten different languages for Tableau
- Set a site-level start page
- Elastic Server on multiple nodes
- SSL connection to TabPy external service
- Configuring the threshold for suspending data-driven alerts
Direct access with TSM CLI

Beginning in the 2019.2 release of Tableau Server, running tsm commands will not require you to enter a password if the following are true:

- The account you are running commands with is a member of the TSM-authorized group, by default, the `tsadmin` group. The Tableau unprivileged user (by default, the `tableau` user) and root account may also run TSM commands.
- You are running commands locally on the Tableau Server that is running the Tableau Server Administration Controller service. By default, the Tableau Server Administration Controller service is installed and configured on the initial node in a distributed deployment.

See Authenticating with tsm CLI.

Local authentication password policy improvements

Beginning in the 2019.2 release of Tableau Server, you can now set password policies for Tableau Server that is configured for local authentication. For example, you can specify password length, expiry, and complexity. Additionally, you can set accounts to be locked out after too many failed sign-in attempts. You can also enable users to reset their own passwords. See Local Authentication.

Select from ten different languages for Tableau

Tableau has added support for two new languages, Chinese (Traditional) and English (United Kingdom). You can now set Tableau to display the user interface in one of ten languages:

- Chinese (Simplified)
- Chinese (Traditional)
- English (United Kingdom)
- English (United States)
- French
- German
- Japanese
Tableau Server on Linux Administrator Guide

- Korean
- Portuguese
- Spanish

For more information, see Language and Locale for Tableau Server.

In addition to supporting two new languages in all Tableau products, the help is also available in these languages. To read the help in your preferred language, navigate to the bottom of the main Tableau Help page, and select the language from the footer.
Set a site-level start page

Beginning in the 2019.2 release of Tableau Server, you can set a start page for all site users, so that they land on a particular page when they sign into Tableau Server. Previously, server administrators could only set a start page at the server-level. See Set the Default Start Page for All Users.

Elastic Server on multiple nodes

Elastic Server is used by Ask Data to index data. You can now run Elastic Server processes on more than one node in a cluster. It is recommended to have an odd number of Elastic Server processes running. For more information, see Tableau Server Processes.

SSL connection to TabPy external service

In previous versions, SSL connections to TabPy were not supported. In Tableau Server 2019.2, you can configure SSL connections. Configuring external services is done with tsm commands at tsm security vizql-extsvc-ssl enable.

Configuring the threshold for suspending data-driven alerts

By default, an alert is suspended after 350 consecutive alert failures. Server administrators can configure the threshold number of alert failures before an alert is suspended. For more information, see Set Up for Data-Driven Alerts.

Added in version 2019.1

Changes to Tableau Help

Updates have been made to help you navigate our content, including a new right-hand menu to quickly scan and select different topics on a page. We've also created content within tabs, to easily show how to complete tasks for web or command line interface.
User

- New features and enhancements related to web authoring and interacting with views on the web are listed in What’s New in Tableau for Users and Web Authoring and Tableau Desktop Feature Comparison. For topics related to web authoring and interacting with views, see Using Tableau on the Web.
- Web authoring auto recovery

Server or site administrator

- Option to install to non-default location on RHEL-distros
- Microservice Container Processes
- Pluggable Authentication Modules (PAM)
- TSM client authentication
- Ask Data and Elastic Server Processes
- Tableau Prep Conductor Server Process
- Node roles for Backgrounder Process
- Tableau Data Management Add-On
- SSL connection to Rserve external service

Web authoring auto recovery

With Tableau Server 2019.1, editing and authoring content on the server will auto-recover in the event of a process failure. As you make changes to content, Tableau Server will intermittently (and silently) save changes. In previous versions, a process failure results in lost changes if you did not save them prior to the failure. In Tableau Server 2019.1, changes you make prior to a failure will be persisted when you reconnect to the content.

By default, Tableau Server will save content after you make five changes to it. Your server admin can adjust this interval with the vizqlserver.NumberOfWorkbookChangesBetweenAutoSaves option using tsm configuration set Options. File size limit (vizqlserver.WorkbookTooLargeToCheckpointSizeKiB) and recovery attempts per session (vizqlserver.RecoveryAttemptLimitPerSession) can also be configured.
Option to install to non-default location on RHEL-distros

Starting with version 2019.1, if you are installing Tableau Server on a RHEL-like distribution of Linux, you can choose to install to a non-default location. The default location is /opt. If you want or need to install Tableau to another location, you can now do so, even when upgrading. Installing to a non-default location is not supported on Ubutu because of limitations of the deb package. For more information about installing Tableau Server, see Install and Initialize TSM. For more information about upgrading, see Upgrade Tableau Server on Linux.

Microservice Container Processes

The 2019.1 release adds some new processes to Tableau Server. Two container processes, Interactive Microservice Container and Non-Interactive Microservice Container now appear on the TSM Status page and in the output of the TSM command tsm status -v. These processes contain microservices and cannot be configured independently by administrators. For more information, see Tableau Server Microservice Containers.

Pluggable Authentication Modules (PAM)

In the 2019.1 release, TSM authentication process uses PAM directly and then falls back to an authentication scheme using substitute user (su) if PAM fails or is not configured with a directory service. This improvement provides a more integration with LDAP directory services. See TSM Authentication.

TSM client authentication

In the 2019.1 release, certificate trust and management for TSM clients (Web UI and CLI) have been updated to make encrypted connections to Tableau Server easier to manage. See Connecting TSM clients.
Ask Data and Elastic Server Processes

With Tableau Server 2019.1, Ask Data and Elastic Server processes have been added to Tableau Server to support the new Ask Data feature. For more information, see Tableau Server Processes.

The Ask Data process is installed and configured automatically. It cannot be configured manually. Logs are stored in the nlp directory. For more information, see Server Log File Locations.

The Elastic Server process is installed and configured automatically. The Elastic Server heap size can be configured by using the elasticsearcher.vmopts TSM configuration option. For more information, see tsm configuration set Options. Only one Elastic Server process can be running and it can optionally be moved to any node in the cluster.

Tableau Prep Conductor Server Process

Tableau Prep Conductor is a new server process that has been added to Tableau Server to support automation of running and managing flows. Tableau Prep Conductor leverages the scheduling and tracking functionality of Tableau Server so you can automate running flows to update the flow output instead of logging into Tableau Prep to manually run individual flows as your data changes. For more details, see Tableau Prep Conductor.

Tableau Prep Conductor is licensed through Tableau Data Management Add-on. For more information, see Licensing Tableau Prep Conductor.

Tableau Data Management Add-On

The Data Management Add-on is a new license that includes Tableau Prep Conductor. For more information, see Licensing Tableau Prep Conductor.

Node roles for Backgrounder Process

A new tsm topology command that gives you the ability to specify the type of tasks that backgrounder can perform has been added. In 2019.1, using the new command, you can
isolate certain nodes to run only flow tasks or run all tasks other than flows. This is useful if you are planning to use Tableau Prep Conductor to publish, run and schedule flows. For more information, see Workload Management through Node Roles and Tableau Prep Conductor.

**SSL connection to Rserve external service**

In previous versions, SSL connections to Rserve and TabPy were not supported. In Tableau Server 2018.3, you can configure SSL connections. In addition, connecting to external services is now enabling, disabling, and configuring is now done with native tsm commands under tsm security.

**Added in version 2018.3**

**Changes to Tableau Help**

Search results within the product help have been expanded to include results for all Tableau products and help articles. Help layout and navigation, such as table of contents placement and breadcrumbs, have been updated to create a more consistent cross-product help experience.

**User**

- New features and enhancements related to web authoring and interacting with views on the web are listed in What's New in Tableau for Users and Web Authoring and Tableau Desktop Feature Comparison. For topics related to web authoring and interacting with views, see Using Tableau on the Web.
- Secure extract data at the row level

**Server or site administrator**

- Materialize calculations in extracts already published to Tableau Server
- Improved content browsing (in-release beta)
- Support for Content Security Policy
Tableau Server on Linux Administrator Guide

- Improved support for LDAP
- SSL connection to Rserve external service
- SAML now supports PKCS#8 key files
- Improved key and secrets refresh process
- Backup without compression (added in version 2018.3.2)

Materialize calculations in extracts already published to Tableau Server

If you need to retroactively materialize calculations in extracts that have already been published to Tableau Server, you can use tabcmd to refresh the published extract with the `--addcalculations` option. For more information about materializing calculations in extracts, see Materialize Calculations in Your Extracts in the Tableau User Help. For more information about the tabcmd command option, see refreshextracts workbook-name or data-source-name.

Secure extract data at the row level

Secure your extract data at the row level using existing approaches for implementing row-level security in Tableau. For more information, see Restrict Access at the Data Row Level in the Tableau User Help.

Improved content browsing (in-release beta)

With Tableau Online and Tableau Server 2018.3, you and your web users can get a sneak preview of our reorganized content browsing experience. The new experience lets you see multiple content types—views, workbooks, data sources, flows, projects—in one grid or list. For example, you can now see all content inside a particular project and its sub-projects.

This new experience intends to make content discovery more intuitive, and it’s the first in a set of planned changes that are aimed at improving self-service and governance of your Tableau content.

To try the new browsing experience, enable the site-level setting we’ve provided for it. For more information, see Preview the Updated Content Browsing Experience.

After you enable the new experience, encourage your users to sign in to our Early Feedback site, share their thoughts with us, and see our regular development updates.
Support for Content Security Policy

Tableau Server supports the Content Security Policy (CSP) standard. CSP is intended to be an additional layer of security against cross-site scripting and other malicious web-based attacks. CSP is implemented as a HTTP response header that allows you to specify where external resources, such as scripts and images, can be safely loaded from.

See Content Security Policy.

Improved support for LDAP

Tableau Server 2018.3 includes improved support for LDAP, such as connection pooling, more flexible root configuration, and commas in object class names. See the new topic, LDAP Configuration Reference.

Connections to external services support SSL

In previous versions, SSL connections to Rserve and TabPy were not supported. In Tableau Server 2018.3, you can configure SSL connections. In addition, connecting to external services is now enabling, disabling, and configuring is now done with native tsm commands under tsm security.

SAML now supports PKCS#8 key files

You can now use password-protected key files with SAML. See Configure Server-Wide SAML. The passphrase is encrypted at rest along with other server secrets. See Manage Server Secrets.

Improved key and secrets refresh process

In previous versions of Tableau Server, refreshing server secrets, keys, and tokens required that admins manually copied the keystore to each node in distributed deployments.

In Tableau Server 2018.3, the keystore is distributed to all nodes when you run the tsm security regenerate-internal-tokens command.
Backup without compression (added in version 2018.3.2)

Tableau Server version 2018.3.2 introduced a new option for the tsm maintenance backup command that allows you to disable compression when doing the backup. This can be beneficial for customers who have backups that take a long time. Using the option speeds up the time for the backup, but results in a larger backup file. If using this command option in a multi-node installation, we strongly recommend you have a File Store instance configured on your initial node to reduce the need to backup across nodes. To learn more about TSM backup, see tsm maintenance backup.

**Added in version 2018.2**

**User**

- Include colleagues on data conversations with @mentions

New features and enhancements related to web authoring and interacting with views on the web are listed in What's New in Tableau for Users and Web Authoring and Tableau Desktop Feature Comparison. For topics related to web authoring and interacting with views, see Using Tableau on the Web.

**Server or site administrator**

- Create a support case from the TSM UI or CLI
- Manage Dashboard Extensions in Tableau Server
- Automated daemon port configuration
- Amazon Linux 2 now supported

Include colleagues on data conversations with @mentions

To engage people in a conversation about compelling data you find in a view, @mention them in comments. As you begin to type a name, matching people with Tableau accounts at
your organization appear in a list you can choose from. For more information, see Comment on Views in Tableau User Help.

Site administrators can disable @mentions in the General section of site settings.

Create a support case from the TSM UI or CLI

You can take a log file snapshot and send it to Tableau Support directly from the Tableau Services Manager UI or CLI. To learn how, see Log File Snapshots (Archive Logs)

Manage Dashboard Extensions in Tableau Server

Dashboard extensions are web applications that provide users new ways to interact with data from other applications directly in Tableau. You can control the dashboard extensions that you allow to run in Tableau. You can manage an extension's access to data and can block extensions to prevent them from running. To learn more about controlling extensions, see Manage Dashboard Extensions in Tableau Server.

Automated daemon port configuration

In previous versions of Tableau Server on Linux, configuring the licensing daemon port was a required, manual step for multinode deployments. In the 2018.2 version, this process happens automatically. Tableau Server on Linux will automatically detect and configure the appropriate ports for the licensing daemon.
Amazon Linux 2 now supported

Beginning with version 2018.2, Amazon Linux 2 (AL2) is now a supported distro for Tableau Server on Linux. For details on all supported distros, see Before you install...

Added in version 2018.1

User

• Resize headers on the web
• Streamline discussion threads by deleting comments

New features and enhancements related to web authoring and interacting with views on the web are listed in What’s New in Tableau for Users and Web Authoring and Tableau Desktop Feature Comparison. For topics related to web authoring and interacting with views, see Using Tableau on the Web.

Server or site administrator

• New user-based term licenses are now available
• New and renamed site roles are now available
• Improved support for managing web data connectors

Resize headers on the web

When editing a view on the web, you can resize the width of row headers and the height of column headers.
Streamline discussion threads by deleting comments

If a comment on a view is unnecessary or inaccurate, users can quickly delete it by clicking the X in the upper-right corner. For more information, see Comment on Views in Tableau User Help.

To disable the ability to delete comments, change the tsm configuration set option, viz-portal.commenting.delete_enabled, to false.

New user-based term licenses are now available

Tableau offers multiple types of user-based term licenses that grant a range of web authoring and other capabilities at various price points, providing the flexibility for organizations to pay for the data analysis and data visualization capabilities that each type of user in their organization needs. To learn more, see User-based licenses.

New and renamed site roles are now available

Starting in the version 2018.1 release, new and updated site roles are now available. To learn about these site roles, see Set Users’ Site Roles. To learn about changes in the REST API to accommodate new site roles, see What’s New in the REST API.

Improved support for managing web data connectors

Tableau Server now provides improved support for web data connectors (WDCs):
A new set of tsm data-access commands are now available to help you manage WDCs.

You can also manage WDCs using the new web-data-connector-settings Entity.

Added in version 10.5

User

- Extracts use the new .hyper format
- Changes to the way values are computed
- Filtered search hidden by default

New features and enhancements related to web authoring and interacting with views on the web are listed in What's New in Tableau for Users and Web Authoring and Tableau Desktop Feature Comparison. For topics related to web authoring and interacting with views, see Using Tableau on the Web.

Server or site administrator

- Optimized server performance for data alerts
- Nested projects
- Tableau Data Engine with Hyper
- Tableau Services Manager
- Hot topology support
- Lightweight Directory Access Protocol (LDAP) support

Optimized server performance for data alerts

To improve overall performance, Tableau Server now rechecks failing data alerts only when it next checks data conditions for all alerts. You can revert to the previous setting, which
checked failing alerts every 5 minutes, by changing the `tsm configuration set` option, `dataAlerts.retryFailedAlertsAfterCheckInterval`, to `false`.

Nested projects

You can now create project hierarchies when you want to create content segments within a larger area. For example, create a project for each department, say Operations. In each of those top-level projects, separate content based on how the audience uses it, say connecting to certified data versus in-progress collaboration. For more information, see the following articles:

- Use Projects to Manage Content Access
- **Navigate projects** in the article Navigate Tableau on the Web

Filtered search hidden by default

Filtered search options are now located on the right side of pages, and are hidden by default. The toggle is a dark icon ( ) when the Filters pane is open and a light icon ( ) when the Filters pane is closed. For details on searching, see Search Content.

To use filtered search, click .
Tableau Data Engine with Hyper

Hyper is Tableau’s new in-memory Data Engine technology optimized to bring faster extract and query performance, and increased scalability. You can now analyze large data sets faster and faster extract creation. For more information, see Tableau Server Data Engine.

Extracts use the new .hyper format

Beginning with version 10.5, new extracts use the new .hyper format. Extracts in the new format take advantage of the improved data engine, which supports the same analytical and query performance as the data engine before it, but for even larger extracts. Similarly, when an extract task is performed on a .tde extract using version 10.5, the extract is upgraded to a .hyper extract. For more information, see Extract Upgrade to .hyper Format.

Changes to the way values are computed

To improve data source efficiency and scalability, and produce results that are consistent with commercial databases, values in your data source can be computed differently in version 10.5. In some cases, these changes can cause differences with the values and marks in...
your view between version 10.4 (and earlier) and version 10.5 (and later). This applies to extract data sources, and can apply to multi-connection data sources, data sources that use live connections to file-based data, data sources that connect to Google Sheets data, cloud-based data sources, extract-only data sources, and WDC data sources. For more information, see Changes to values and marks in the view in the Tableau User Help.

Tableau Services Manager

Tableau Server version 10.5 features our new web-based server configuration and management application, Tableau Services Manager (TSM). To learn more about TSM, see Tableau Services Manager Overview.

Hot topology support

Tableau Server includes support for hot topology changes. This allows you to change the number of Backgrounder and VizQL processes running on nodes that already have these processes - without the need to restart Tableau Server. To learn more about these and other Tableau Server processes, see Tableau Server Processes.

Lightweight Directory Access Protocol (LDAP) support

Tableau Server now supports Lightweight Directory Access Protocol (LDAP). To learn more about LDAP support in Tableau Server, see Identity Store
What's Changed - Things to Know Before You Upgrade

This article lists important changes in Tableau Server, beginning with version 10.5. Version 10.5 was the first version of Tableau Server for Linux. This list is cumulative. If you are upgrading from an early version (for example, 10.1), read the list of changes for every version between your current version and the version you are upgrading to. If you are upgrading from Tableau Server on Windows to Tableau Server on Linux, read both versions of "What's Changed."

Looking for Tableau Server on Windows? See What's Changed - Things to Know Before You Upgrade.

What Changed in Version 2019.4

Version 2019.4 includes the following changes you should know about before upgrading.


New tabcmd Installer for Windows

The 2019.4 release of Tableau Server features a newly rewritten Setup program for installing tabcmd on Windows.

This update changes some of the command line options. If you have automated the installation of tabcmd, be sure to review the current command line options for the tabcmd Setup program. For more information, see tabcmd and Install Switches and Properties for tabcmd
Updates to Tableau Server Job Management

The Job management page has several updates and new options including the following:

- The Jobs page now includes **Average Run** and **Average Queue** time information for each job.

- Improved job cancellation experience. Server and Site admins can now enter customized notes as well as pick recipients of the e-mail when cancelling a job.

- More information is provided in the job detail dialog box, such as: the last time the job was run successfully, and the name of the job creator.

- Canceling flow run jobs is now supported. Previously, in Tableau Server version 2019.3, only extract refreshes and subscriptions were supported.

- New **Task Type** has been added: Encryption jobs.

- Failed jobs now includes jobs that have been suspended. Suspended jobs are a sub-status of the failed jobs and have their own icon to distinguish them from other failed jobs.

For more information, see Managing Jobs in Tableau Server.

What Changed in Version 2019.3

Version 2019.3 includes the following changes you should know about before upgrading.

Default cipher suite no longer includes default support for old ciphers

The 2019.3 version of Tableau Server no longer includes default support for Triple-DES, IDEA, and CAMELIA ciphers for SSL connections. These ciphers are no longer considered adequately secure. In the previous versions of the Security Hardening Checklist, we included a recommendation to disable the Triple-DES cipher, which was enabled by default.

Depending on your environment, you may need to make additional configurations before or after upgrading:

- If you disabled the Triple-DES cipher as documented in the Checklist, then we recommend running the following commands before you upgrade to 2019.3:

  ```bash
  tsm configuration set -k ssl.ciphersuite -d
  tsm pending-changes apply
  ```

  This command reverts Tableau Server to the 2019.2 (and older) default cipher suite configuration. After you have upgraded to 2019.3, Triple-DES, IDEA, and CAMELIA will be removed from the default supported ciphers.

  Do not run the command if you made other changes to the `ssl_ciphersuite` key.

- If you did not change the `ssl_ciphersuite` key, then Setup will update the key value to the new default as part of the upgrade process. The new default value for the `ssl_ciphersuite` key is

  ```text
  ```

- If your users still require support for Triple-DES, then we recommend upgrading to more recent browsers that support more secure ciphers. If that is not an option, then you can run the following commands after you have upgraded. The first command sets the `ssl_ciphersuite` key to the pre-2019.3 configuration, which supports Triple-DES:
tsm configuration set -k ssl.ciphersuite -v "HIGH:MEDIUM:!aNULL:!MD5:!RC4"

tsm pending-changes apply

Tableau Server Service Manager has been removed

The Tableau Server Service Manager (tabsvc) has been removed. Instead, systemd manages the Tableau business processes that were previously managed by tabsvc:

- Gateway
- Repository & Cluster Controller
- File Store
- Application Service
- SAML Service
- Backgrounder
- Data Server
- VizQL Server
- Data Engine
- Cache Server
- Search & Browse

You can use the following scripts to start and stop the business services that tabsvc managed in previous versions:

start-administrative-services

stop-administrative-services

The scripts are located at /opt/tableau/tableau_server-/packages/scripts.<version>/.

Usage data is sent to Tableau

Beginning with Tableau Server 2019.3, server usage data is sent to Tableau by default. This is the case for new installs and upgrades. You have the ability to opt out during install-
To disable the sending of server usage data:

- If you are installing Tableau Server for the first time, you can disable this feature during the configuration of initial node settings. For details, see Configure Initial Node Settings.

- If you are upgrading Tableau Server, the upgrade-tsm script will let you know about the option and how to disable it.

- If you want to have installed or upgraded Tableau Server and want to disable the sending of server usage data, you can do this from the TSM Maintenance page, or from the command line. For details, see Server Usage Data.

Upgrade without needing to provide a password

Beginning with Tableau Server 2019.3, when you upgrade from version 2019.2 or later, the upgrade-tsm script no longer requires you to specify a user and password. Version 2019.3 and later take advantage of the Direct access with TSM CLI feature and uses the account you have logged in as to authenticate the upgrade. You can override this behavior with the --user option, to specify an administrative user. When you do this, you can also provide a password using the --password option, or enter a password when prompted. Upgrades from versions earlier than 2019.2.0 will still prompt you for a password.

Some tsm command options are no longer configurable

Beginning with Tableau Server 2019.3, the following options are enabled on install and are no longer configurable:

- backgrounder.enable_sort_jobs_by_job_rank
- backgrounder.enable_task_run_time_and_job_rank

What Changed in Version 2019.2
Version 2019.2 includes the following changes you should know about before upgrading.


**Minimum hardware requirements have changed**

Beginning with version 2019.2, Tableau Server requires a minimum of 4 cores and 16 GB of RAM in order to install. The installation program will check that the computer you are installing on satisfies these minimums before you can install. If the computer does not meet these hardware minimums, Tableau cannot be installed. For details, see Minimum Hardware Requirements and Recommendations for Tableau Server. For more information about this change, see this Tableau blog post: [Updated machine requirements for Tableau Server trials](https://www.tableau.com/support/minimum-requirements).

**Upgrade requires at least 15% free disk space on each node**

Upgrading to version 2019.2 includes a version update to the PostGRES database that Tableau uses internally. This update means that a backup of the database is done before upgrading, and that requires a minimum of 15% free space on each node.

**"Referrer-Policy" HTTP header added**

Beginning in 2019.2, Tableau Server includes the ability to configure Referrer-Policy HTTP header behavior. This policy is enabled with a default behavior that will include the origin URL for all "secure as" connections (policy no-referrer-when-downgrade). In previous versions, the Referrer-Policy header was not included in responses sent by Tableau Server. This change is not likely to impact user scenarios for most browsers. However, we recommend testing browser-based user scenarios after you upgrade to 2019.2. In the event a user scenario is effected, you can disable or change this header behavior. See [HTTP Response Headers](https://www.tableau.com/support/http-response-headers).
Map-related internet addresses have changed

Beginning in 2019.2, Tableau Server needs access using port 443 to two new map locations:

- mapsconfig.tableau.com
- api.mapbox.com

These replace the address used by previous versions: maps.tableau.com. For details on configuring Tableau Server for proper internet access, see Configuring Proxies for Tableau Server.

What Changed in Version 2019.1

Version 2019.1 includes the following changes you should know about before upgrading.


Install to non-default locations (RHEL-like distributions only)

Beginning with version 2019.1, on RHEL-like distributions, you can install Tableau Server to a location other than the default /opt. This is not supported on Ubutu due to limitations of the deb package. For details, see install instructions in Install and Initialize TSM or upgrade instructions in Upgrade Tableau Server on Linux.

High-resolution thumbnails

In version 2019.1 and later, workbooks and views use higher resolution thumbnails. When you upgrade to Tableau Server version 2019.1 or later, the Backgrounder runs the Enqueue Thumbnails Upgrade job and the Upgrade Thumbnails job, which refreshes thumbnails for any views that have not been edited or published since Tableau version 2018.2 and converts them from low resolution thumbnail images (192 x192 pixels) to high resolution thumbnail images (300 x 300 pixels), as shown in the figure below.
When the Upgrade Thumbnails job refreshes the thumbnail images, the modified date of the view is changed to match the thumbnail update date. The Upgrade Thumbnails job does not change the modified date of workbooks when it updates the contained view’s thumbnails.

Thumbnails may appear blurry until the Upgrade Thumbnails job is complete. Depending on the number of workbooks you have, this could a few minutes to a few hours. You may also notice that thumbnail images may appear in low resolution for up to a week after upgrading to Tableau Server 2019.1. For more information, see Upgrade Thumbnails Job.

Project card images

In version 2019.1 and later, project card images will not appear on the project card unless they are added as the last item in the project description and encapsulated with ! (exclamation marks). For more information, see Add a Project Image.

What Changed in Version 2018.3

Version 2018.3 includes some changes you should know about before upgrading.

Floating worksheet captions become transparent

To make a floating caption visible again, content authors will need to change the caption background to a color.

Changes to some Data Engine tsm configuration set options

Two new configuration tsm set configuration options have been added: `hyper.hard_concurrent_query_thread_limit` and `hyper.soft_concurrent_query_thread_limit`

The new options replace the `hyper.num_job_worker_threads` and `hyper.num_task_worker_threads` options available in Tableau Server versions 2018.3 and earlier. The `hyper.num_job_worker_threads` and `hyper.num_task_worker_threads` have been deprecated and will no longer work in 2019.1 and later.

SSL Off-loading and SAML

Beginning with version 2018.3, Tableau Server validates SAML response message returned from the IdP. If your organization terminates SSL connections from the IdP at a proxy server before sending the authentication request to Tableau Server, then users may be unable to login with SAML after you upgrade from 2018.2.

In this scenario, SSL is "off-loaded" at the proxy server, which means the https request is terminated at the proxy server and then forwarded to Tableau Server over http. Since SSL is off-loaded at the proxy, Tableau Server will validate with the protocol that it receives (http), but the IdP response is formatted with https, so validation will fail unless your proxy server includes the X-Forwarded-Proto header set to https. See Configuring Proxies for Tableau Server.

What Changed in Version 2018.2

Version 2018.2 includes some significant changes you should know about before upgrading.

The following sections summarize the significant changes to Tableau Server 2018.2 and provide links for additional information.

- Updates to Tableau Server Apache Logs
- The Read Only site role is now deprecated
- The tabcmd initialuser command prompts for password
- New ports added

Moving the repository in a two-node cluster not supported

Updates to Tableau Server Apache Logs

Apache log files stored in /logs/httpd now contain records with several formatting improvements, including the following:

- Server names are shown according to the UseCanonicalName setting, which makes it easier to distinguish between nodes in a multi-node server deployment.
- Date and time are now tracked in a single date-and-time string that can be readily recognized as a single value by a variety of data analysis tools, including Tableau Desktop.
- Time zones are now easier to parse
  - On Linux, the numerical offset from UTC is provided.
  - On Windows, the name of the timezone is enclosed in quotes

Date, time and time zone formatting changes are illustrated by an example log entry from Tableau Server version 2018.1 and earlier releases:

::1 - - 2018-05-09 08:51:48.872 GMT Daylight Time 80 "POST /vizql/w/Re-
gional/v/Obesity/sessions/A971A92AF77A4F4A9813065BDCEB24D6-0:0/-
For purposes of comparison, here is a log entry from Tableau Server version 2018.2:

10.210.24.3 127.0.0.1 - 2018-05-09T16:07:58.120 "GMT Daylight Time" 80 "POST /vizql/w/Super-store/v/Overview/bootstrapSession/sessions/185CCDC854A44765BB0298E93B403879-0:3 HTTP/1.1" "-" 200 136026 "784" 2370951 WvMOzgLfhzh9kFWO@ow2gAAA1Y

The Read Only site role is now deprecated

The Read Only site role that was introduced in Tableau Server version 2018.1 is now deprecated. Any users assigned to the Read Only site role in version 2018.1 are reassigned to the Viewer site role after an upgrade to version 2018.2. To learn more about site roles, see Set Users' Site Roles.

The tabcmd initialuser command prompts for password

In previous versions, the tabcmd initialuser command required a --password parameter to set the initial administrator password. In the tabcmd included with this version of Tableau Server, you can leave the --password parameter out of the command to be prompted in the shell instead. See initialuser.

New ports added

We've added new service ports in 2018.2.

Automatic port assignment

For installations where automatic port assignment is enabled (this is recommended and the default) there should be no impact to upgrading. The new ports will be automatically selected
on upgrade.

**Manual port assignment**

If you have disabled automatic port assignment, *you will need to manually specify the new ports in your configuration before upgrading*. For more information about manually assigning ports, see Manual port assignment.

**Ports added in 2018.2**

<table>
<thead>
<tr>
<th>Port names</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>You must define each of these for every node if you disable dynamic port mapping.</td>
<td></td>
</tr>
<tr>
<td>hyper.port</td>
<td>Data engine primary port.</td>
</tr>
<tr>
<td>hyper.connection.port</td>
<td>Data engine connection port.</td>
</tr>
<tr>
<td>licenseservice.vendor_dae-mon.port</td>
<td>License service vendor daemon port.</td>
</tr>
<tr>
<td>saml.service.port</td>
<td>SAML service port.</td>
</tr>
<tr>
<td>tabadminagent.columbo.port</td>
<td>Administration Agent service discovery port</td>
</tr>
<tr>
<td>tabadminagent.filetransfer.port</td>
<td>TSM Agent file transfer port.</td>
</tr>
<tr>
<td>vizportal_0.microservice.extensions.port</td>
<td>Application server extensions port.</td>
</tr>
<tr>
<td>vizportal_0.monolith_grpc.port</td>
<td>Application server GRPC port.</td>
</tr>
</tbody>
</table>

**What Changed in Version 2018.1**

Version 2018.1 includes some changes you should know about before upgrading.

The following sections summarize the significant changes to Tableau Server 2018.1 and provide links for additional information.

- Removed tabadmin command options
- Updated tsm commands
- Updated upgrade-tsm script

Removed tabadmin command options

The previously deprecated --license and --administrator options are no longer available when using the `tabcmd createsiteusers`, `tabcmd createusers` and `tabcmd syncgroup` commands. These are breaking changes for any scripts that still use the --license or --administrator options. To update your scripts, use the --role option with these commands to specify the license levels or site roles. For details, see `tabcmd` Commands.

Updated tsm commands

As of version 2018.1 we have added or expanded the following tsm commands:

- `tsm maintenace cleanup`—The tsm maintenance cleanup command was added in version 10.5.1. For version 2018.1 it has been enhanced with additional options to give you more flexibility. For details, see tsm maintenance cleanup.
- `tsm reset` —Added in version 2018.1, this command allows you to reset the initial administrative user. For details, see tsm reset.

Updated upgrade-tsm script

As of version 2018.1 the upgrade-tsm script you run when you upgrade from a previous version now requires you to include the name of a TSM administrator when you run it. For
example:

```bash
sudo /opt/tableau/tableau_server/packages/scripts.<version_code>/upgrade-tsm -u <tsm_system_admin> -p <tsm_system_admin_password> --accepteula
```

where `tsm_system_admin` is a user with administrative permissions on the initial node computer.

### What Changed in Version 10.5

Version 10.5 includes some changes you should know about before upgrading.


The following sections summarize the significant changes to Tableau Server 10.5 and provide links for additional information.

- Tableau Server configuration changes
- Changes to extracts
- Version compatibility
- Viewer (can publish) and Unlicensed (can publish) site roles removed
- Nested projects and content navigation
- Nested projects and tabcmd
- Filtered search hidden by default
Hyper powers the new Data Engine in Tableau Server 10.5 and introduces the changes described below. For more information, see Tableau Server Data Engine.

Beginning with version 10.5, Tableau Installer automatically installs Data Engine when you install file store, Vizportal, VizQLServer, data server, or backgrounder and cannot be installed separately. Every node that has an instance of one of these processes will also have a single instance of the data engine process. You can no longer configure more than one instance of data engine on a node. To learn more about Tableau Server Processes see Tableau Server Processes.

Previously, in versions 10.4 and earlier, for distributed installations, the backgrounder process launched temporary instances of data engine (tdeserver64.exe) process during extract creation or refreshes. In Tableau Server 10.5, a single instance of data engine is installed automatically when you install backgrounder. The backgrounder process uses this single instance of data engine that is installed on the same node. For more information on configuration recommendations, see Distributed Installation Recommendations.

Upgrade may affect your current configuration for a multi-node cluster, especially if you have certain processes isolated on a separate node. For more information, see Tableau Server Data Engine.

Changes to extracts

Beginning with version 10.5, new extracts use the new .hyper format. The improved data engine can read both the .tde and .hyper file formats, so the latest version of Tableau Server can continue to view and read workbooks that are using .tde extracts.

Existing extracts will not automatically be upgraded to the new .hyper format during the Tableau Server or Tableau Desktop upgrade process. Only when certain extract tasks are performed on the .tde extract, the extract is upgraded to the .hyper format. For example, an automatic or scheduled refresh will upgrade a .tde extract to a .hyper extract.
For more information on extract upgrades, see Extract Upgrade to .hyper Format.
For recommendation on how to handle automated or schedule refresh tasks before you upgrade, see Prepare for the Upgrade.

NOTE: Use the Extract API 2.0 to create (.hyper) extracts, which requires Tableau 10.5 and later. If you are using Tableau 10.4 and earlier, continue to use the Tableau SDK to create .tde extracts. The Tableau SDK is no longer being maintained. For more information, see Extract API 2.0.

Version compatibility

Upgraded extracts are not backward compatible. After extracts are upgraded from .tde to .hyper format, they cannot be opened or refreshed in Tableau Desktop 10.4 and earlier. Tableau recommends that Tableau Server and all Tableau Desktops in your organization be on the same version and be upgraded to 10.5 on the same schedule. For more information on specific extract compatibility scenarios, see Impact of extract upgrade

Viewer (can publish) and Unlicensed (can publish) site roles removed

The site roles **Viewer (can publish)** and **Unlicensed (can publish)** have been deprecated and are no longer available. Users who have either of these site roles on Tableau Server 10.4 or earlier will be reassigned in Tableau Server 10.5.

<table>
<thead>
<tr>
<th>Deprecated site role</th>
<th>In 10.5 is reassigned to</th>
</tr>
</thead>
<tbody>
<tr>
<td>Viewer (can publish)</td>
<td>Viewer</td>
</tr>
<tr>
<td>Unlicensed (can publish)</td>
<td>Unlicensed</td>
</tr>
</tbody>
</table>

Nested projects and content navigation

You can now create project hierarchies when you want to segment content into sub-sections within a larger area. For example, you might create a project for each department. In each of those top-level projects, you can separate content based on how the audience uses
it, such as connecting to certified data versus in-progress collaboration. For more information, see the following articles:

- **Use Projects to Manage Content Access**

  - **Navigate projects** in the article *Navigate Tableau on the Web*

Nested projects and tabcmd

Using tabcmd, you can specify only a top-level project in a project hierarchy. To automate tasks you want to perform on a sub-project or content in a sub-project, you must use the equivalent Tableau REST API call.

Not all actions are available through the REST API. If you use Server on Windows and want to run an unscheduled extract refresh or append a file to an extract, a possible alternative is to use the *Tableau Data Extract Command Line Utility*.

Filtered search hidden by default

Filtered search options are now located on the right side of pages, and are hidden by default. The toggle is a dark icon ( ) when the Filters pane is open and a light icon ( ) when the Filters pane is closed. For details on searching, see *Search Content*.

To use filtered search, click .
Minimum Hardware Requirements and Recommendations for Tableau Server

The following minimum hardware requirements and recommendations apply to all computers running Tableau Server, including physical hardware and virtual machines (VMs):

- **Minimum requirements for testing and prototyping** are the minimum hardware your computer must have in order to install Tableau Server. If your computer does not meet these requirements, the Setup program will not install Tableau Server. These requirements are appropriate for testing and prototyping, but are not appropriate for most production environments.

- **Minimum recommendations for production** go beyond minimum requirements, and represent the minimum hardware configuration you should use for a production installation of Tableau Server. If your computer meets the minimum requirements but does not meet these recommendations, the Setup program will warn you but you can continue the installation.
In addition, Tableau Server should not be installed on a physical computer or on a VM instance that is also running resource-intensive applications such as databases or application servers.

**Note:** If you install Tableau Server on a computer that meets the minimum requirements but does not have at least 8 cores and 16 GB of system memory, the default number of all processes installed is reduced to one of each process by design. For more information about processes, see Server Process Limits.


Minimum Hardware Requirements for Testing and Prototyping

The computer on which you are installing or upgrading Tableau Server must meet the minimum hardware requirements. If the Setup program determines that your computer does not meet the following requirements, you will not be able to install Tableau Server.

These minimum requirements are appropriate for a computer that you use for prototyping and testing of Tableau Server. They apply to single-node installations and to each computer in a distributed installation.

<table>
<thead>
<tr>
<th>Processor CPU</th>
<th>RAM Free</th>
<th>Disk Space</th>
</tr>
</thead>
<tbody>
<tr>
<td>64-bit 4-core</td>
<td>16 GB</td>
<td>15 GB</td>
</tr>
</tbody>
</table>

**Minimum Hardware Requirements**

**Note:** These minimum requirements are not recommended for use in production environments. For production minimum recommendations, see Minimum Hardware Recommendations.
For the requirements:

- Free disk space is calculated after the Tableau Server Setup program is unzipped. The Setup program uses about 1 GB of space. You may need to allocate additional disk space depending on various factors like whether you will be using extracts. For more information, see Disk Space Requirements.

The core Tableau Server bits must be installed in a directory with at least 15 GB of free disk space. If you attempt to install Tableau Server on a computer that does not have enough space, the Tableau Server package will install, but you will be unable to continue with setup. By default the install location is the /opt directory. You can change the installation path for Tableau Server on RHEL distros. For more information, see Option to install to non-default location on RHEL-distros.

If you plan to make heavy use of extracts then you may need to allocate additional disk space. You can specify a different directory for data (extract) storage during installation. See Install and Initialize TSM.

- Core count is based on "physical" cores. Physical cores can represent actual server hardware or cores on a virtual machine (VM). Hyper-threading is ignored for the purposes of counting cores.

Minimum Hardware Recommendations

For production use, the computer on which you install or upgrade Tableau Server should meet or exceed the minimum hardware recommendations. These recommendations are general. Actual system needs for Tableau Server installations can vary based on many factors, including number of users and the number and size of extracts. If the Setup program determines that your computer does not meet the following recommendations, you will get a warning, but you can continue with the setup process. The minimum recommendations listed here are intended as general guidance. However the recommendations for your environment may vary. For more information, see the Hardware recommendations section of the Recommended Baseline Configurations topic.
**Install Type** | **Processor** | **CPU** | **RAM** | **Free Disk Space**
--- | --- | --- | --- | ---
Single node | 64-bit | 8-core, 2.0 GHz | 32 GB | 50 GB
| | or higher

If you are adding Tableau Prep Conductor to your Tableau Server installation, we recommend you add a second node and dedicate this to running Tableau Server Prep Conductor. This node should have a minimum or 4 cores (8 vCPUs), and 16 GB of RAM.

**Multi-node and enterprise deployments**

Contact Tableau for technical guidance.

Nodes must meet or exceed the minimum hardware recommendations, except:

- Nodes running backgrounder, where 4 cores may be acceptable.
- Dedicated node for Tableau Prep Conductor: Minimum of 4 cores (8 vCPUs), and 16 GB of RAM.

For the recommendations:

- Free disk space is calculated after the Tableau Server Setup program is unzipped. The Setup program uses about 1 GB of space. You may need to allocate additional disk space depending on various factors like whether you will be using extracts. For more information, see Disk Space Requirements.

The core Tableau Server bits must be installed in a directory with at least 15 GB of free disk space. If you attempt to install Tableau Server on a computer that does not have enough space, the Tableau Server package will install, but you will be unable to continue with setup. By default the install location is the `/opt` directory. You can change the installation path for Tableau Server on RHEL distros. For more information, see Option to install to non-default location on RHEL-distros.
If you plan to make heavy use of extracts then you may need to allocate additional disk space. You can specify a different directory for data (extract) storage during installation. See Install and Initialize TSM.

- Core count is based on "physical" cores. Physical cores can represent actual server hardware or cores on a virtual machine (VM). Hyper-threading is ignored for the purposes of counting cores.

- RAM shown is the minimum recommended for a single-node installation. Your installation may function better with more RAM, depending on activity, number of users, and background jobs, for example. For hardware specifications Tableau uses internally for testing scalability, see Hardware Recommendations.

For hardware recommendations for Tableau Server in the cloud, see the following:

- Selecting an AWS Instance Type and Size in the Tableau Server on Linux in the AWS Cloud Administrator Guide

- Selecting a Google Compute Engine Virtual Machine Type and Size in the Tableau Server on Linux in the Google Cloud Platform Administrator Guide

- Selecting a Microsoft Azure Virtual Machine Type and Size in the Tableau Server on Linux in Microsoft Azure Administrator Guide

Prepare for the Upgrade

To properly prepare for a Tableau Server upgrade, gather the following information about your existing installation, key files related to your installation and the upgrade, and complete the pre-upgrade tasks.

Gather custom configuration information

If you are upgrading on the existing hardware your configuration will be preserved, but it's a good practice to collect this configuration information about your existing installation for several reasons: you need this information when you configure a test environment, you need this information if you are migrating to new hardware migration as part of the upgrade, and
you can use the information to confirm that the upgraded Tableau Server is configured as expected if you notice something unexpected after upgrading.

Collect this information and any associated files and save them to a location that is not on any of the Tableau Server computers.

The following list includes examples of the type of information you should gather:

- **TSM Controller certificate** Verify the certificate for the TSM Controller is still valid.

  To verify your TSM Controller SSL certificate's expiration date:

  1. Open a terminal session.

  2. Type the following commands to display the dates when the certificate is valid:

     ```
     openssl s_client -connect <tsm_servername>:8850
     echo | openssl s_client -connect <tsm_servername>:8850 2>/dev/null | openssl x509 -noout -dates
     ```

  3. If the certificate is expired, open a case with our Support team, and they will be happy to help with the upgrade.

- **Customizations** This includes non-default ports, timeout values, custom logo images, and fonts.

- **SMTP** configuration.

  For more information, see Configure SMTP Setup.

- **SSL** configuration and certificates.

  For more information, see Configure SSL for External HTTP Traffic to and from Tableau Server.

- **SAML** configuration, certificates and any IdP metadata files.
For more information, see SAML.

- **Kerberos** configuration.

  For more information, see Configure Kerberos.

- **OpenID** configuration.

  For more information, see Configure Tableau Server for OpenID Connect.

- **Additional node** configurations. Collect the configurations of any additional Tableau Server nodes, including any certificates or other supporting files that you copied to these nodes.

- **Other** values. Note the number of projects, groups, workbooks, views, data sources, and users you have in your production environment. Having this information makes it easy to do a quick check after the upgrade to make sure everything was restored as expected.

Gather the environment configuration

The steps you need to take during the upgrade process depend on whether you're installing the upgrade on the same hardware or you're migrating to new hardware. Upgrading on the same hardware is straightforward and requires a minimum of manual steps (the steps you need to take depend on what version you are upgrading from, what version you are migrating to, and whether or not your existing installation is in the default location). Migrating to new hardware requires you to manually restore your Tableau Server data and reconfigure your settings after you install the new version.

Copy the setup files to a local file share

You will need the following setup files before you upgrade Tableau Server:
Tableau Server on Linux Administrator Guide

- The installer for your existing version of Tableau Server.

You might not need the installer. However, we recommend that you have it available in case there's a problem during the upgrade. That way you can use the setup program and your server backup to restore your installation to its pre-upgrade state.

**Note:** If you do not have the installer for your existing version, you can download it from the Alternate Downloads Site. Save the installer in a safe location that is not part of your production or test version of Tableau Server. You will need the Setup program if you need to go back to your existing version after upgrading.

- The setup program for the new version of Tableau Server.

**Update tabcmd**

A new version of tabcmd is released with every release of Tableau Server. If you install tabcmd on computers that are not part of your Tableau Server installation, you need to update tabcmd on those computers. For more information, see tabcmd.

**Back up Tableau Server data**

We recommend that you make a backup of your installation of Tableau Server before beginning the upgrade process. This provides data that you'll need to set up a test version of the upgraded environment. It also lets you recover if the upgrade process fails.

**Notes:**

- Backups from versions of Tableau Server older than version 9.x cannot be used with version 2018.2.x or later. If you are running an older version of Server, you may need to do an intermediate upgrade to version 9.x through 10.x before upgrading to version 2018.2 or later.

- We recommend you disable subscriptions and scheduling in your production
environment immediately before taking the backup, and reenable them after the backup is complete. Doing this will help avoid having your users receive duplicate subscriptions and email messages when you restore your backup in your test environment.

- The full backup can take a while if you have a large installation or a lot of extracts.

- Any changes made between the time you took the backup and the time you do the upgrade are lost because they aren't included in the backup.

- You can only restore from a backup that has the same type of identity store as the running server. For example, a backup from a server using local authentication can be restored to a Tableau Server initialized with local authentication, but a backup from a server using Active Directory authentication cannot be restored to a server initialized with local authentication.

For more information, see Remove Unneeded Files and Back up Tableau Server data.

Generate asset keys file before uninstalling Tableau Server for Windows 2018.1.x or earlier

If you are upgrading from Tableau Server on Windows version 2018.1.x or earlier, and you generated a custom asset key, save a copy of the asset_keys.yml file before you uninstall your earlier version. Store this file in a safe place that is not part of your Tableau Server installation (a network share for example).

In most cases you will not need this file. The upgrade process updates asset keys automatically when an upgrade is successful. You only need this file if the upgrade fails and you need to install a new version of Tableau Server and manually restore your backup file. In this case, you also need to specify the asset keys file when doing the restore.

Locating an existing asset keys file

If you generated a custom asset key file, you will have an asset_keys.yml file in the data directory. By default this is:

\ProgramData\Tableau\Tableau Server\data\tabsvc\config
If you do not see an `asset_keys.yml` file, you can skip this.

Restoring a pre-2018.2 Tableau Server for Windows backup along with asset keys

To manually restore a backup from Tableau Servers for Windows 2018.1.x or earlier, and include asset keys, use `tsm`, for example:

```
  tsm maintenance restore --file <backup> --ak <asset_key_file>
```

For more information, see `tsm maintenance restore`.

Check your product maintenance status

If you attempt to upgrade a Tableau Server installation that has a product key with expired maintenance, your upgraded Tableau Server will be unlicensed and the upgrade may fail. Before upgrading, make sure that the server’s maintenance hasn’t expired.

If your maintenance has expired, select the product key and then click **Refresh**. If this does not update the maintenance date, check the Tableau Customer Portal for a replacement Tableau Server product key with a later maintenance date. For any questions or concerns, please contact Tableau Technical Support. Reactivating the product key will be part of the upgrade process.

For more information, see Activate and Register Tableau Server. If your server doesn’t have internet access, see Activate Tableau Server Offline - Adding a License.

Disable your extract refreshes

Beginning in 10.5, extracts use .hyper format instead of .tde. Once the file format has been upgraded to .hyper, it cannot be reverted back to .tde format and cannot be opened in Tableau Desktop version earlier than 10.5. For more information, see Extract Upgrade to .hyper Format.

The following tasks on Tableau Server will upgrade a .tde extract to a .hyper extract:

- Manual refresh
- A scheduled full or incremental extract refresh
- Automated refresh tasks that are performed through tabcmd, Extract API 2.0, or through Extract Command Line Utility

- Automated append data to an extract using tabcmd or using the Extract Command Line Utility

**Note:** We recommend that you disable any automated tasks and schedules that result in an extract upgrade. If you have desktop versions that have not yet been upgraded to 10.5, your desktop users will not be able to open the upgraded .hyper extracts. You may also want to first test a few extracts manually after doing the server upgrade, and then reenable any schedules or automated refresh tasks.

For more information about extract refresh scheduling, see Enable Extract Refresh Scheduling and Failure Notification.

Plan your strategy if you use virtual machines (VMs)

If you run Tableau Server on VMs, either locally, or in the cloud, be aware of the potential for complications related to licensing. If you are simply upgrading Tableau Server on the VM, you do not need to take any extra action related to licensing. If you plan to clone the VM to create either a new production or test environment to upgrade, you need to deactivate any Tableau Server licenses before cloning. If you do not do this, the new VM environment can end up with an untrusted license, and any attempts to upgrade will fail. You may also end up hitting the maximum number of activations for the licenses.

To avoid issues with licensing on VMs, deactivate all Tableau licenses before cloning a VM or allowing it to be permanently shut down.

**Test the Upgrade**

The best way to learn what impact a Tableau Server upgrade will have to your current environment is to test it. Knowing how an upgrade will affect your users and your server helps you plan and communicate before the actual upgrade, ensuring that your users will not be caught by surprise.
If you have a Tableau Server test environment this is a great place to test out the upgrade.

We recommend the following sequence for testing a Tableau Server upgrade:

1. Prepare a test environment
2. Upgrade the test environment
3. Confirm that existing functionality works
4. Performance and user acceptance testing
5. Test new features
6. Communicate about the upgrade

Prepare a test environment

To start, create a test environment that mirrors your production environment as closely as possible. The closer your test environment is to the actual environment you will be upgrading, the more accurate a representation you will have of how the upgrade will impact you. This includes identical or similar hardware and operating systems, as well as the same authentication options and network access.

When you've got a test computer or virtual machine ready, follow these steps for creating a test environment.

1. On the existing production environment, create a backup of Tableau Server using the `tsm maintenance backup` command.

   For more information, see Create a pre-upgrade backup.

2. On your test environment, install a copy of the same version of Tableau Server as you have in your production environment.
3. Restore your existing database data using the `tsm maintenance restore` command.

   For more information, see Restore from a backup.

4. Manually replicate your existing Tableau Server configuration.

   You need to manually configure certain aspects of your environment because when you restore the Tableau database it doesn’t include configuration details and customizations.

Upgrade the test environment

Follow the appropriate steps for upgrading the test environment, based on your environment:

- Upgrade Tableau Server on Linux
- Upgrade Tableau Server on Linux from 10.5

Confirm that everything works as expected

After you have the new version of Tableau Server installed and configured in your test environment, you are ready to test. You should test basic functionality, along with any special aspects of server that your organization relies on. For example, if there are key subscriptions that your organization relies on, make sure that you test those.

These are some areas of testing to consider:

- **Server processes.** Sign in to Tableau Server as a server administrator, and then open the Server Status page to confirm that all services and processes are running as expected (including on all additional nodes if this is a distributed installation).
• **User access.** Confirm that Tableau Server users can sign in. Test your normal user sign in process. Have some of your users participate in the testing to make sure they are able to sign in as expected, and that they can get to the same content that they have access to in your production environment.

• **Publishing workbooks and data sources.** Have users publish workbooks and data sources from Tableau Desktop to make sure this goes as you expect.

• **Viewing published workbooks.** Have users who are familiar with the content try to view published workbooks to make sure they appear as expected. Test views embedded in web pages (for example, in SharePoint pages).

• **Subscriptions and extract refreshes.** Manually run some extract refreshes to confirm that they complete successfully. Run some key scheduled extract refreshes to confirm that they complete as expected.

• **Permissions.** Confirm that permissions are still set as expected for users and content.

• **Command-line utilities and APIs.** If applicable, test the command line utilities (tsm and tabcmd) and programmatic access via APIs.

Performance and user acceptance testing

Use tools like Tabjolt, Replayer, and Scout to do performance and user acceptance testing on your test environment. For more information about these and other performance testing tools, see Performance Monitoring Tools. If you have workbooks that have calculations and use extracts as the data source, we recommend that you test the performance of these workbooks before you upgrade. For more information, see Test Workbook Performance.

Test new features

Take a look at the new features that come with the version you are upgrading to, and at any features that were added between the version you currently have and the new version. Think about how to help your users understand the benefits of the features that apply to your environment.
Communicate about the upgrade

The best way to make an upgrade go smoothly is by letting your organization know ahead of time about the upgrade and how it might impact them. If you've had users help test, take advantage of their experience by having them help communicate the changes they saw while testing. You can also provide user access to the test environment if there are key people who should see the upgraded version before the actual upgrade.

Test Workbook Performance

In most cases, you can expect to see similar or better performance after upgrading your Tableau Server from versions 10.4 or earlier to 10.5 or later. However, for workbooks that use complex calculations, you may see performance issues after you upgrade. This article describes how to test performance and troubleshoot these performance issues before you upgrade your production Tableau Server from versions 10.4 and earlier to 10.5 or later.

If you have already upgraded your production Tableau Server to 10.5 or later, and you are seeing performance issues for workbooks that have calculations and use extracts as the data source, see Troubleshoot Performance of Workbooks with Calculations.

Since Linux was first released in 10.5, this only applies when you are migrating from Tableau Server versions 10.4 or earlier on Windows to Linux 10.5 or later.

To make sure your workbook performance is same or better after your upgrade to Tableau 10.5 or later, we strongly recommend that you set up a test environment and do performance testing before you upgrade your production Tableau Server.

Use the following sequence of steps for testing workbook performance:

1. Set up your test environment
2. Capture performance metrics and analyze
3. Troubleshoot performance issues
4. Capture the conclusions and results
Set up your test environment

1. Create a test environment that mirrors your production environment as closely as possible.

   Follow the steps described in Migrate Tableau Server from Windows to Linux.

2. Disable any automated or scheduled extract refresh schedules.

Capture performance metrics and analyze

1. Run Tabjolt or any other tool of your choice to capture performance metrics on your current version of Tableau Server to set a baseline. Use workbooks that have calculations and use extracts as the data source for testing. For more information about how install and use Tabjolt, see the Tabjolt Installation Guide.

2. Perform a full backup to capture the metrics from the first step.

3. Upgrade your test environment to 10.5.3 or later. We are recommending 10.5.3 as the minimum version since the settings you need to enable in order to resolve performance issues later on are only available in 10.5.3 or later versions of Tableau.

4. **Checkpoint:** Run Tabjolt to compare workbook performance before and after upgrade.

   If you see performance degradation after upgrade at this step, it may be due to other Tableau Server issues and might require further investigations. For more information, see Record and Analyze Workbook Performance.

   If you see no difference in performance before and after upgrade, continue to the next step.

5. Select a subset of workbooks and run a full refresh of the extracts. Doing a full refresh will upgrade the extracts from .tde to .hyper.

6. **Checkpoint:** Run the tool again to compare the performance of your workbooks between step 4 and step 6 which is essentially before and after doing extract
refreshes. Specifically, check for workbooks with slower response times or failure to load.

Troubleshoot performance issues

If you find that you are unable to test due to performance issues, or you find that VizQL server is consuming all or most of the available memory on the machine while viewing them, follow steps 1 through 3.

For workbooks that load successfully, but have slower response after upgrade, follow steps 4 and 5.

1. Enable the following setting that checks the number of query nodes required for Tableau to generate the workbook and displays an error when the set limit is exceeded. This helps prevent VizQL process from consuming the memory resources on the machine that might result in critical Server issues:

   Use TSM CLI to run the following commands:

   ```
   tsm configuration set -k native_api.node_limit_checker_pre_rewrite_disable -v false
   tsm pending-changes apply
   ```

   **Note:** This setting is only available in Tableau Server 10.5.3 or later. While enabling this setting improves the reliability and stability of Tableau Server, you may see errors on a very small percentage of workbooks with when you enable this setting. For example, the Tableau team saw errors on 0.01% of the workbooks with this setting enabled on Tableau Online.

2. Test the workbooks by viewing or interacting with them. You should see an error message informing you about exceeding the node limits on workbooks that have performance issues - The error message might say something like: "Logical Query tree
3. Use Tableau Desktop to troubleshoot your workbook performance. Run this command to enable the node limit check in your Tableau Desktop:

```bash
tableau.exe -DNodeLimitCheckerPreRewriteDisable=false
```

**Note:** This setting is only available in Tableau Desktop 10.5.3 or later.

In Tableau Desktop, download and open the workbooks that displayed the error message about high node limits in step 2. Try optimizing the performance of workbooks using methods described in the Best Practices for Creating Calculations in Tableau topic. If successful, then publish them to Tableau Server. If you are still experiencing performance issues, continue to the next step.

4. For workbooks that load successfully, but the overall response times is slower after upgrade, materialize calculations for your extract:

Using Tabcmd:

```bash
tabcmd refreshextracts --workbook "My Workbook" --addcalculations
```

For more information, Compute Calculations Now option.

5. If you have a large number of workbooks that fall under the above category (overall response times is slower after upgrade), consider enabling the server wide setting to retain materialized calculations for all extracts when they are upgraded from .tde to .hyper:

Using TSM CLI:

```bash
tsm configuration set -k native_api.preserve_calculations_on_hyper_refresh_conversion -v true

calling tsm pending-changes apply
```
Considerations for turning on this setting:

Turn this setting on only if majority of your workbooks require calculations to be materialized. There are resource and time costs associated with this setting, as described below:

- When this setting is set to true, the extract file size will increase, affecting the overall disk space.
- Extracts with materialized calculations take a longer time to refresh than extracts without materialized calculations.

This setting only affects extracts that are not yet upgraded to .hyper. Extracts already upgraded to .hyper before turning on this setting will not have the calculations materialized. This setting will also not affect workbooks and extracts published from Tableau Desktop, or any new extract refreshes published to Tableau Server that do not use the Compute Calculations Now option.

Note: This setting is only available in Tableau Server 10.5.3 or later.

6. Enable the extract refresh schedules and monitor the performance of your workbooks to make sure that your workbooks are performing as expected.

Capture the conclusions and results from your testing

Make a note of the settings and steps that helped resolve the performance issues to use them to prevent performance degradation when you upgrade your Production Tableau Server.

For example, if during your testing, you found that you needed to set the native_api.preserve_calculations_on_hyper_refresh_conversion setting to true, remember to disable your extract refreshes just before you upgrade your production Tableau Server.
After upgrade, set the `native_api.preserve_calculations_on_hyper_refresh_conversion` setting to true, and then re-enable your extract refreshes after you have confirmed that you are not experiencing slow response times after upgrade.

Perform the Upgrade

After you've completed the tasks in Prepare for the Upgrade, you're ready to upgrade your existing Tableau Server installation to version 2019.4

As a best practice, you should always make a backup of your Tableau Server data before upgrading, and save this backup in a safe place that is not part of your Tableau installation. This backup preserves your Tableau data if something unexpected happens during the upgrade, or if you need to go back to your previous version of Tableau Server. For more information, see Back up Tableau Server data.

Upgrade Tableau Server on Linux

To upgrade Tableau Server you install the new version while your existing version is running, then run an upgrade script to switch over to the new version. This reduces the amount of downtime for users because the server continues to run while you install the new version. You only stop the server to run the upgrade script.

**Important:** Upgrading Tableau Server on Linux from version 10.5.x (10.5.0 or later) requires special steps that are documented separately. For instructions for how to upgrade from 10.5.x, see Upgrade Tableau Server on Linux from 10.5.

General notes about upgrading:

All upgrades:

- **Backup**—Create a backup. You are responsible for creating a backup of your existing Tableau Server data before upgrading. We strongly recommend creating a backup before upgrading as this provides you with an up-to-date version of your data you can
use if you need to go back to your earlier version of Tableau Server. For more information on properly preparing for an upgrade, see Prepare for the Upgrade.

- **Licenses**—

  - **Trial product keys**—You cannot directly upgrade a server that is running a trial product key from one major version to another (for example, from version 10.5.x to version 2018.x). To upgrade from one major version to another with a trial product key, install the new version of Tableau Server on a separate machine, activate a trial key there, and restore a backup from your existing version. For more information, see Confirm licensing requirements.

  - **Expired maintenance**—You cannot upgrade a server with a product key whose maintenance has expired or expires on a date earlier than the release date of the version you are upgrading to. If you attempt to upgrade a server in this state, your server will be unlicensed and the upgrade may fail. If your maintenance has expired or is too old, refresh your product key before you run the upgrade script to upgrade to your new version. If refreshing the key does not update maintenance to a valid date, check in the Tableau Customer Portal for a Tableau Server key with current maintenance. For details on refreshing the product key, see Refresh Maintenance Date for the Product Key.

- **Ports**—Before upgrading, read and understand the requirements for ports: Tableau Services Manager Ports.

- **Install while running**—When you upgrade Tableau Server, you install the new version before uninstalling the old version. The upgrade process will leave the old version in place. You can uninstall the old version after the upgrade is complete to free up disk space, though this is not required.

- **Upgrade script**—To complete the upgrade and switch to the new version, after installing the new version on all nodes in your cluster, run the upgrade script upgrade-tsm on the initial node. The script is installed in the
Tableau Server must be stopped to complete the upgrade. If Tableau Server is not stopped when you run the script, the script will let you know, and will offer to stop the server. You can also choose to stop the server before running the script using the `tsm stop` command.

- **Multiple directories**—After an upgrade, you will see multiple versions of Tableau Server directories in the `/packages` directory. When the upgrade is completed successfully, you can delete the directories related to the older version you upgraded from. You can identify these by the version code in the directory name. For example, `scripts.<version_code>`. For more information about version codes, see View Server Version.

Multi-node upgrades:

- **Node installation**—Run the installer on all nodes in a cluster.

The steps below describe the general process for an upgrade of Tableau Server on Linux. Upgrades from beta versions of Tableau Server on Linux are not supported. If you are a Tableau Server on Windows user and want to switch to Tableau Server on Linux, see Migrate Tableau Server from Windows to Linux.


As a best practice you should always make a full backup before upgrading any version of Tableau Server. Save this backup file to a location that is not part of your server installation. Creating and saving a backup preserves your pre-upgrade data and configuration and gives you a rollback option in the event of issues during the upgrade. For more information, see Back up Tableau Server data.
When upgrading Tableau Server on Linux, you do not need to uninstall the previous version. After upgrading you can uninstall the package for the previous version if you choose to do so. One reason to do this is to free up disk space used by the older version. To understand the difference between removing and uninstalling Tableau Server on Linux, see Uninstall Tableau Server.

Upgrading Tableau Server requires a stop and start the server as part of the upgrade process. During this stop/restart Tableau Server is unavailable.

Upgrading will also apply any pending changes. If you have pending changes you do not want applied, you can discard those changes using the `tsm pending-changes discard` command. To see what changes might be pending, use the `tsm pending-changes list` command.

Upgrading Tableau Server on Linux

Follow these steps on each node in your cluster to upgrade Tableau Server.

1. On each node in your cluster:
   a. Copy the new Tableau Server .rpm or .deb package to a location accessible from the computer you are upgrading.
      
      If you are upgrading a distributed deployment of Tableau Server, then copy the .rpm or .deb package to each node in the cluster or to a location accessible from each node.
   b. Log on as a user with sudo access to the computer you are upgrading.
   c. Navigate to the directory where you copied the .rpm or .deb Tableau Server package.
   d. Use the package manager to install the Tableau Server package.
On RHEL-like distributions, including CentOS, you have the option to install Tableau to a non-default location.

- **Default location**—To install to the default location (/opt/tableau/tableau_server), run the following commands, where `<version>` is formatted as major-minor-maintenance (ex: 2019-2-5):

  ```
sudo yum update

sudo yum install tableau-server-<version>.x86_64.rpm
  ```

- **Non-default location**—To install to a non-default location, you must use `rpm -i`. You will also need to install all dependent packages. See the note below.

  Run the following command:

  ```
sudo rpm -i --prefix /preferred/install/path tableau-server.rpm
  ```

**Note:** When you use `yum` to install Tableau Server, all dependent packages are automatically downloaded and installed. This is the preferred method for installing Tableau. If you want to install to a non-default location, or your organization does not allow you to use `yum` and you must install using `rpm -i`, you must also install all dependent packages separately. For information about installing dependent packages, see Installing Tableau Server on an Air-Gapped Computer Running Linux.

On Ubuntu and Debian, run the following commands, where `<version>` is formatted as major-minor-maintenance (ex: 2019-2-5):
2. After you have installed the new package on every node in your cluster, run the upgrade script on the initial node to complete the upgrade. If Tableau Server is not stopped when you run the script, the script will let you know, and will offer to stop the server. You can also choose to stop the server before running the script using the `tsm stop` command. Tableau Server must be stopped to complete the upgrade.

3. With Tableau Server stopped, run the upgrade script on the initial node. Do not run the script on any additional nodes. The options you need to include depend on the version you are upgrading to:

   - Version 2019.3 or later:

     ```
sudo /opt/tableau/tableau_server-/packages/scripts.<version_code>/upgrade-tsm --accepteula
     ```

     where `<version_code>` is the long form of new version you are upgrading to, for example `scripts.20183.18.1128.2033`.

     Starting with version 2019.3.0, when you upgrade from 2019.2.x or later, the script runs using the account you are logged in with. If you are prompted, enter your password. For more information, see Upgrade without needing to provide a password. You can specify a different user with administrative permissions using the `-u` option and specifying a user with administrative permissions on the computer where the initial node is installed. You will be prompted for the password for the administrative user.

   - Version 2018.1 through version 2019.2.x:
sudo /opt/tableau/tableau_server-/packages/scripts.<version_code>/upgrade-tsm -u <sys-

tem_admin> --accepteula

where <version_code> is the long form of new version you are upgrading
to, for example scripts.20183.18.1128.2033, and <system_admin> is a user with administrative permissions on the computer where the initial node is installed. You will be prompted for the password for the administrative user.

The -u option was added as of 2018.1. For more information, see Updated
upgrade-tsm script.

To see all the options available for the upgrade-tsm script, use the -h option. For example:

sudo /opt/tableau/tableau_server/packages/scripts.<version_-
code>/upgrade-tsm -h

4. After the upgrade is completed, exit the terminal session on the initial node and log in
again. This ensures that your session will be using the updated TSM version.

5. Start Tableau Server:

    tsm start

If you have purchased the Tableau Data Management Add-on or the Tableau Server Man-
agement Add-on you can activate them once you have completed your upgrade. For more
information, see the following topics:

• About Tableau Server Management Add-on
• Use the Data Management Add-on

Related topics
• Common Tableau Server Upgrade Issues
Upgrade Tableau Server on Linux from 10.5

When you upgrade Tableau Server on Linux from version 10.5, you need to take several unique steps to complete the upgrade. These are necessary because of a change made after version 10.5.0 released, related to sudo privileges. For more information, see System User, sudo Privileges, and systemd. You only need to do these extra steps once, during the upgrade to 2018.1 or later. This topic describes how to upgrade from version 10.5.0 or 10.5.x (10.5.1 or later) to version 2018.1 or later.

If you attempt to upgrade from 10.5.0 or 10.5.x without following these instructions, warnings are displayed and the upgrade is canceled. You will not break your existing Tableau Server installation, but you cannot continue the upgrade.

To identify the version of your installation, see View Server Version.

Follow these steps to upgrade from 10.5.0:

1. Upgrade to 10.5.x—If you are running version 10.5.0, you must first upgrade to 10.5.x (10.5.1 or higher) by installing 10.5.x and running the upgrade-tsm script in the 10.5.x scripts directory on your initial node.

2. Install 2018.x or later—With 10.5.x installed and running as expected, install 2018.x or later, but do not upgrade to this version yet.

3. Run TSM commands—Use TSM to stop the server and run three additional commands.

4. Migrate 10.5.x to single user—Run the migration script in the new version (2018.x or later) scripts directory. Do this on every node in your cluster.

5. Upgrade to 2018.x or later—Upgrade Tableau Server by running the upgrade-tsm script from the new version scripts directory on your initial node.
Upgrade to 10.5.x

If you are running version 10.5.0 of Tableau Server on Linux, the first step you need to take is to upgrade to a later version of 10.5. Beginning with version 10.5.1 changes were made that are needed in order to upgrade to 2018.1 or later. (If you are already on a version of 10.5 that is higher than 10.5.0, you can skip to the Install 2018.x or later step.)

To upgrade from 10.5.0 to a later version of 10.5:

1. On each node in your cluster:
   a. Copy the Tableau Server version 10.5.x .rpm or .deb package to location accessible from the computer you are upgrading.
      If you are upgrading a distributed deployment of Tableau Server, copy the .rpm or .deb package to each node in the cluster or to a location accessible from each node.
   b. Log on as a user with sudo access to the computer you are upgrading.
   c. Navigate to the directory where you copied the .rpm or .deb Tableau Server package.
   d. Use the package manager to install the Tableau Server package.
      - On RHEL-like distributions, including CentOS, run the following command:
        
        ```
        sudo yum install tableau-server-<version>.x86_64.rpm
        ```
      - On Ubuntu, run the following commands:
        
        ```
        sudo gdebi -n tableau-server-<version>_amd64.deb
        ```

2. Stop Tableau Server. If you are upgrading a cluster, do this after you have installed the new package on every node in your cluster.
tsm stop

3. With Tableau Server stopped, run the following command on your initial node. Do not run this command on any additional nodes:

```
sudo /opt/tableau/tableau_server-/packages/scripts.<version_code>/upgrade-tsm --accepteula
```

where `<version>` is the 10.5.x version you are upgrading to.

To see all the options available for the `upgrade-tsm` script, use the `-h` option. For example:

```
upgrade-tsm -h
```

4. After the upgrade is completed, ensure your session is using the updated TSM version by doing one of the following:

- Use the source command:

  ```
  source /etc/profile.d/tableau_server.sh
  ```

- Exit the terminal session on the initial node and log in again.

5. Start Tableau Server:

```
tsm start
```

Install 2018.x or later

Install the new Tableau Server package but do not upgrade to this version yet. Before you do so, you need to run several commands and a migration script. You can install the new version package without stopping the server. When you install the new package you are copying the software to your computer but not changing anything about the currently running version.

To install the new version package, on each node in your cluster:
1. Copy the Tableau Server .rpm or .deb package to location accessible from the computer you are upgrading.

   If you are upgrading a distributed deployment of Tableau Server, then copy the .rpm or .deb package to each node in the cluster or to a location accessible from each node.

2. Log on as a user with sudo access to the computer you are upgrading.

3. Navigate to the directory where you copied the .rpm or .deb Tableau Server package.

4. Use the package manager to install the Tableau Server package.

   - On RHEL-like distributions, including CentOS, run the following command:
     ```
     sudo yum install tableau-server-<version>.x86_64.rpm
     ```
   - On Ubuntu, run the following commands:
     ```
     sudo gdebi -n tableau-server-<version>_amd64.deb
     ```

Run TSM commands

Using version 10.5.x of Tableau Server that is installed and running:

1. Stop the server:

   ```
   tsm stop
   ```

2. Run these three commands:

   ```
   tsm configuration set -k service.linux.privileged_user -v 'tableau'

   tsm configuration set -k install.username -v 'tableau'

   tsm pending-changes apply
   ```
where 'tableau' is the user name you specified with the initialize-tsm --unprivileged-user option when you first installed 10.5.x. If you did not specify a user, the default is 'tableau'.

Migrate 10.5.x to single user

Run this script from the 2018.x or later scripts directory:

```bash
sudo /opt/tableau/tableau_server/packages/scripts.<version_code>/migrate-to-single-user
```

where `<version_code>` is the long form of your new version number.

**Important:** If you have a multi-node installation, you must run this script on every node in your cluster.

At this point Tableau Server is running 10.5.x but configured to work with a single user. This is an interim stage. You should complete the upgrade to version 2018.x or later before using Tableau.

Upgrade to 2018.x or later

After completing the above steps:

1. With Tableau Server stopped, run the upgrade script on the initial node. Do not run the script on any additional nodes. The options you need to include depend on the version you are upgrading to:

   - Version 2019.3 or later:

     ```bash
     sudo /opt/tableau/tableau_server-/packages/scripts.<version_code>/upgrade-tsm --accepteula
     ```

     where `<version_code>` is the long form of new version you are upgrading to, for example `scripts.20183.18.1128.2033`. 
Starting with version 2019.3.0, when you upgrade from 2019.2.x or later, the script runs using the account you are logged in with. If you are prompted, enter your password. For more information, see Upgrade without needing to provide a password. You can specify a different user with administrative permissions using the \texttt{-u} option and specifying a user with administrative permissions on the computer where the initial node is installed. You will be prompted for the password for the administrative user.

- **Version 2018.1 through version 2019.2.x:**

  \begin{verbatim}
  sudo /opt/tableau/tableau_server-/packages/scripts.<version_code>/upgrade-tsm -u <system_admin> --accepteula
  \end{verbatim}

  where \texttt{<version_code>} is the long form of new version you are upgrading to, for example \texttt{scripts.20183.18.1128.2033}, and \texttt{<system_admin>} is a user with administrative permissions on the computer where the initial node is installed. You will be prompted for the password for the administrative user.

  The \texttt{-u} option was added as of 2018.1. For more information, see Updated upgrade-tsm script.

  To see all the options available for the \texttt{upgrade-tsm} script, use the \texttt{-h} option. For example:

  \begin{verbatim}
  sudo /opt/tableau/tableau_server/packages/scripts.<version_code>/upgrade-tsm -h
  \end{verbatim}

  2. After the upgrade is completed, ensure your session is using the updated TSM version by doing one of the following:

    - Use the source command:

      \begin{verbatim}
      source /etc/profile.d/tableau_server.sh
      \end{verbatim}
Exit the terminal session on the initial node and log in again.

3. Start Tableau Server:

```bash
tsm start
```

When desired, you can remove Tableau Server on Linux version 10.5 from your server. Unlike most other programs that run on Linux, previous Tableau Server versions are not automatically removed as part of a successful upgrade. To learn more, see Remove Tableau Server from Your Computer.

Related topics

- Common Tableau Server Upgrade Issues

Troubleshoot Tableau Server Install and Upgrade

Follow the suggestions in this topic to resolve common issues with Tableau Server. For additional troubleshooting steps based on process status viewed on the Status page, see Troubleshoot Server Processes.

General Troubleshooting Steps

Many Tableau Server issues can be addressed with some basic steps:

1. Make sure there is enough disk space on each computer running Tableau Server. Limited disk space can cause a failure to install, a failure to upgrade, or problems running Tableau Server.

2. Restart Tableau Server. Issues related to processes not fully started can be resolved by restarting Tableau Server in a controlled way. To restart Tableau Server, use the `tsm restart` command. This will stop all the processes associated with Tableau Server and then restart them.

3. Reindex Tableau Server. Issues related to indexing can be resolved by reindexing
Tableau Server on Linux Administrator Guide

Tableau Server. To reindex Tableau Server, use the `tsm maintenance reindex-search` command. For more information, see Reindexing Tableau Server Search & Browse below.

4. Restart the computer on which Tableau Server is running. Some issues, such as those related to data source connectivity, can be resolved by restarting the server computer.

Common Tableau Server Install Issues

Installation logs location

The install log, `app-install.log`, is located in `/var/opt/tableau/tableau_server/logs`.

The upgrade log, `app-upgrade.log`, is located in `/var/opt/tableau/tableau_server/logs`.

Unable to log into TSM or Tableau Server (sign in screen redisplays after entering credentials)

Using Internet Explorer or Edge, if you enter your credentials into the TSM or Tableau Server sign-in screen and the page redisplays without signing you in, verify that the hostname or domain in your URL does not include an underscore (_). If the hostname or domain of the Tableau Server computer includes an underscore (_), Internet Explorer or Edge browsers will not set a cookie, so the page will redisplay without signing you in. To work around this, use "localhost" or the IP address of the computer in the URL. For example: `https://localhost:8850`.

For more information, see the Tableau Knowledge Base.

Multiple install attempts fail

If you attempt to install Tableau Server and the install fails, any subsequent installation attempts are likely to fail unless you run the `tableau-server-obliterate` script to
clean Tableau off the computer.

A failed install attempt can leave the computer in a state that causes subsequent attempts to also fail with errors that don't seem directly related to a previous install attempt. One possible error is:

Enabling and starting all services
+ services=(appzookeeper* tabadmincontroller* tabsvc*
licenseservice* fnplicenseservice* tabadminagent* clientfileservice*)
+ systemctl_user enable appzookeeper_0.service
'tabadmincontroller*' 'tabsvc*' 'licenseservice*' fnplicense-
service_0.service 'tabadminagent*' 'clientfileservice*'
++ id -ru a_tabadminpoc
+ local unprivileged_uid=222954
+ su -l a_tabadminpoc -c 'XDG_RUNTIME_DIR=/run/user/222954 systemd --user enable appzookeeper_0.service tabadmincontroller*
tabsvc* licenseservice* fnplicenseservice_0.service tabadmin-
agent* clientfileservice*'

Failed to execute operation: No such file or directory

To fix this problem, run the tableau-server-obliterate script to clean up any left over remnants of the previous install attempt and then restart the computer. For more information, see Running the tableau-server-obliterate script.

**Important:** If you created a backup of Tableau (<file>.tsbak) you want to keep (for example, to restore to your new installation), copy that file to a safe location on another computer to guarantee it is not removed when you clean up your Tableau computer.

Install fails due to hardware requirements

Tableau Server cannot install if the computer you are installing on does not meet the minimum hardware requirements. The requirements apply to all computers on which you are
installing Tableau Server. For details on minimum hardware requirements, see Minimum Hardware Requirements and Recommendations for Tableau Server.

Common Tableau Server Upgrade Issues

Upgrade logs location
The upgrade log, `app-upgrade.log`, is located in `/var/opt/tableau/tableau_server/logs`.

Upgrade multi-node, initializing additional node fails with "Enter your credentials again" error

If you attempt to initialize an additional node when upgrading Tableau Server and see this error:

Enter your credentials again. The credentials you enter must provide administrative access to the computer where you generated the configuration file.

this is an indication that the node is unable to connect to or communicate with the initial node. This can happen for multiple reasons:

- The credentials you entered are not valid or you mistyped them. The credentials must be for a user who has administrative permissions on the computer where Tableau Server was first installed. You do not need to use the credentials of the user who created the bootstrap file but doing so will ensure you are using valid credentials.

- The local firewall of the computer you are trying to add is not allowing communication to the initial node. For more information, see Local firewall configuration.

- The nodes are on different subnets. In distributed installations, all nodes should be installed on the same subnet. For more information, see Distributed Requirements.

Upgrading fails due to lack of disk space

If there is not enough disk space for the Tableau Server Setup program to run and do the upgrade, the installation will fail. The amount of disk space required will depend on the size of
your repository database and the number and size of your extracts. As a part of upgrading to version 9.0, the Setup program migrates extracts to the new File Store and this takes space.

To free up disk space:

1. Create a log archive snapshot using the `tsm maintenance ziplogs` command.  
   After you create the ziplogs file, save it to a safe location that is not part of your Tableau Server installation.

2. Clean up unnecessary files using the `tsm maintenance cleanup` command.  
   For more information, see Remove Unneeded Files.

Troubleshooting connections to TSM

Starting Tableau Server

Tableau Server cannot determine if it fully started

In some instances Tableau Server may report that it could not determine if all components started properly on startup. A message displays: "Unable to determine if all components of the service started properly."

If you see this message after starting, verify that Tableau Server is running as expected by using a `tsm status -v` command.

If the status shows as running ("Status: RUNNING"), then the server successfully started and you can ignore the message. If the status is DEGRADED or STOPPED, see "Tableau Server doesn't start" in the next section.

Tableau Server doesn't start

If Tableau Server does not start or is running in a degraded state, run the `tsm restart` command from a command prompt. This will shut down any processes that are running, and restart Tableau Server.
Reindexing Tableau Server Search & Browse

Problems that can be solved by rebuilding Search & Browse index

Symptoms of an index that needs to be rebuilt include:

- A blank list of sites when a user attempts to log in
- A blank list of projects when a user tries to select a project
- Missing content (workbooks, views, dashboards)
- Unexpected or inaccurate alerts (for example, an "refresh failed" alert on a workbook that does not include an extract)

If you see any of these behaviors, reset and rebuild the Search & Browse index using the `tsm maintenance reset-searchserver` command.

Activating Tableau Server

Tableau Server license activation fails

In some instances Tableau Server license activation may fail. Error messages can range from a very generic one:

- An error has occurred

To more specific messages:

- Function `flxActCommonLicSpcPopulateFromTS` returned error 50030, 71521,

- No license found for 'Tableau Server'

To resolve this issue, try these solutions in the order listed:

Confirm you can access the licensing server

The Tableau licensing service was moved to a new data center on October 6, 2018. This means any environments that required special configuration (static IP safe listing for example) to access licensing.tableau.com or licensing.tableau.com will need to be updated before you can activate, refresh, or deactivate a Tableau product key.
Tableau Server needs to connection to the following internet locations for licensing purposes:

- licensing.tableau.com:443
- atr.licensing.tableau.com:443
- s.ss2.us
- ocsp.rootg2.amazonsct.com
- ocsp.rootca1.amazonsct.com
- ocsp.sca1b.amazonsct.com
- crt.sca1b.amazonsct.com
- crt.rootca1.amazonsct.com
- ocsp.sca0a.amazonsct.com
- crt.sca0a.amazonsct.com
- ocsp.sca1a.amazonsct.com
- crt.sca1a.amazonsct.com
- ocsp.sca2a.amazonsct.com
- crt.sca2a.amazonsct.com
- ocsp.sca3a.amazonsct.com
- crt.sca3a.amazonsct.com
- ocsp.sca4a.amazonsct.com
- crt.sca4a.amazonsct.com
- crl.rootca1.amazonsct.com
Requests to the above domains may be on port 80 or 443.

Verify the date and time

Verify the date and time on the initial Tableau Server computer is correct. If the clock is set to a time and date earlier than the current date, Tableau Server cannot be activated.

Force the product key to be read again

1. On the initial Tableau Server computer, sign in as administrator and open a command prompt.

2. Change to the Tableau Server bin directory. By default this is:

   /opt/tableau/tableau_server/packages/bin.<version_code>/

3. Type the following commands:

   tsm stop

   ./lmreread

   tsm start

Send the contents of trusted storage to Tableau Support

If FlexNet Licensing Services is installed and running but you’re still seeing an error, there might be a problem with the Tableau product key information. To resolve this issue, complete the following steps to create a file of the key information located in trusted storage.

1. On the initial Tableau Server computer, sign in as administrator and open a command prompt.

2. Type the following command:
serveractutil -view > <machine_name>-LicResults.txt

This creates the <machine_name>-LicResults.txt file in your current directory. If you don't have write permissions for that location and see an error, change to a location where you do have permission to create a file and run the command again.

3. Contact Tableau Support (http://www.tableau.com/support/request) and include the <machine_name>-LicResults.txt file that you created.

Uninstall Tableau Server

Do not uninstall Tableau before upgrading. For details on upgrading, see Upgrade Tableau Server on Linux.

You can have multiple versions of Tableau Server installed at the same time. This allows you to run most of an upgrade while an existing version is running, and reduces downtime and impact to users. Once you have upgraded, you can uninstall your previous version. Doing this frees up disk space. You do not have to uninstall the previous version.

This article explains how to uninstall previous versions, after you've upgraded to a newer version.

Uninstalling and completely removing Tableau Server

There are two primary "uninstall" scenarios that Tableau Server on Linux supports:

- **Uninstall Tableau Server**: After you upgrade to a new version of Tableau Server you can uninstall your previous version to free up disk space. Continue reading for information about uninstalling Tableau.

- **Remove Tableau Server**: If you want to completely remove Tableau Server from a computer, you can use a script provided by Tableau to remove Tableau Server and all related files. This removes all data as well as server components, so should only
be done if you know you want to reset the computer to a pre-Tableau state. You might need to do this if Technical Support recommends this step when troubleshooting an installation problem. We recommend you create a backup of your data before removing Tableau. Save the backup file to a safe location on a computer that is not part of your Tableau installation. Completely remove Tableau Server without uninstalling any version first. The script will uninstall all existing versions found on the computer. If you have already uninstalled your existing version and now want to completely remove Tableau, you can find the script to do so in a temporary location. For more details, see Remove Tableau Server from Your Computer.

Uninstall a Tableau Server package

Use this procedure to free up disk space by uninstalling packages for previous Tableau Server versions after you have upgraded to a newer version of Tableau Server.

1. Look at the environment.bash file to confirm which version of Tableau Server is currently in use. At a command prompt, type:

   ```bash
grep TABLEAU_SERVER_DATA_DIR_VERSION /etc/-
   opt/tableau/tableau_server/environment.bash
   ```

2. Determine which versions of the Tableau Server package are installed on your computer.
   - On RHEL-like distributions, including CentOS, run the following command:
     ```bash
     yum list installed tableau-server"*"
     ```
   - On Ubuntu and Debian, run the following command:
     ```bash
     apt list --installed tableau-server"*"
     ```

3. Remove the Tableau Server package with your package manager.
   - On RHEL-like distributions, including CentOS, run the following command:
     ```bash
     sudo yum remove tableau-server-<version>.x86_64
     ```
On Ubuntu and Debian, run the following commands:

```bash
sudo apt-get purge tableau-server-<version>_amd64
```

Reinstall a Tableau Server package that was accidentally uninstalled

Do not uninstall the package for your currently running version of Tableau Server. Doing so will make the server unusable. To completely remove Tableau Server and all its files, see Remove Tableau Server from Your Computer.

When you uninstall the Tableau Server package for the current instance of Tableau Server the following operations run:

- All files under /opt/tableau/tableau_server are removed. These files are the unmodified installation files.
- Tableau Server services are stopped and disabled
- Service files for all Tableau Server services are persisted
- Data files are left in place

If you accidentally uninstall the package for your currently running version of Tableau Server, follow this procedure to correct the situation.

To reinstall after uninstalled the running instance of Tableau Server:

1. Reinstall the package for the version you accidentally uninstalled.
2. Run `initialize-tsm`.

Remove Tableau Server from Your Computer

**Warning:** The steps below completely remove Tableau Server on Linux, and delete users and groups created by `initialize-tsm`, all related data, configuration information, and logs. This includes any files in `/tmp` or `/var/tmp` that are owned by users configured in `/etc/opt/tableau/tableau_server/environment.bash` as
privileged and unprivileged users (by default, tsmagent and tableau). Tableau Server licenses are also deactivated, unless you omit the -l option when running the command shown below.

If you want to uninstall a particular Tableau Server package to free up disk space (after upgrading, for example), see Uninstall Tableau Server.

As part of the regular installation of Tableau Server, a script is installed that provides you a way to completely remove Tableau and all associated files from your computer. This is something you would only do if you did not care about your Tableau data, configuration, or log files, or if you are working with Tableau Technical Support and need to reinstall Tableau Server after an installation or upgrade attempt fails. The obliterate script will not remove any drivers you installed separately, even those you installed to use with Tableau Server.

The tableau-server-obliterate script is intended for when you want to completely remove Tableau Server from your computer. You might want to do this for a couple of different reasons:

- You no longer want Tableau Server installed on the computer. Use the tableau-server-obliterate script to remove Tableau Server completely. If this is the case, you can include a -l option to deactivate the Tableau Server license.

- Troubleshooting Tableau Server installation problems—If you run into issues installing Tableau, you may need to use the tableau-server-obliterate script to completely remove Tableau Server from your computer before reinstalling. Doing this will clean up any older settings or states (such as the /etc/opt/tableau/tableau_server/environment.bash file) and allow you to reinstall on a "clean" computer. If you are doing this, you can leave off the -l option to preserve licensing information on the computer. When you omit the -l option, you will not need to activate your license when you reinstall Tableau Server.
The script will remove all log files, so if you run into an issue that you want to troubleshoot, or if Tableau Support requests log files for a support case, you should zip up the log files before running the script. For details on gathering log files, see Work with Log Files.

If you have a backup file you want to save (if you will be reinstalling Tableau for example), copy it to a safe location on a separate computer.

What tableau-server-obliterate does

The intent of the `tableau-server-obliterate` script is to completely remove Tableau Server from your computer. This includes files, system settings and configurations, and, if you specify, licensing information.

**Note:** An older version of `tableau-server-obliterate` may miss files from new versions of Tableau Server. As a best practice you should always run the obliterate script from the most recent version of Tableau you installed.

When you run `tableau-server-obliterate`, the following steps are taken:

- Uninstall is run for all installed versions of Tableau Server (yum erase or apt-get remove).
- All contents of the data directory is removed (by default `/var/opt/tableau/tableau_server`).
- All semaphores and shared memory segments are deleted.
- All temp files owned by the "tableau" user are deleted from `/tmp` and `/var/tmp`.
- All users and groups created during install are deleted.
- `/etc/opt/tableau` is deleted.
- Trusted certificates are removed from `/etc/pki/ca-trust` and `/usr/share/ca-certificates/tableau`.
- Configuration files are removed from:
  - `/etc/sysctl.d/99-tableau-server.conf`
  - `/etc/profile.d/tableau_server*`
  - `/etc/security/limits.d/99-tableau_server*`
Tableau Server on Linux Administrator Guide

- /etc/systemd/logind.conf.d/tableau_server*
- /usr/share/bash-completion/completions/tsm
- etc/bash_completion.d/tsm
- /usr/share/bash-completion/completions/tabcmd
- /etc/bash_completion.d/tabcmd
- /run/tableau
- /usr/lib/tmpfiles.d/tableau-server.conf

- All server licenses (except trial licenses) are deactivated if you use the -l option.

Running the tableau-server-obliterate script

You can completely remove Tableau Server from a computer, either preserving the licensing information, or removing the licensing information along with all other aspects of Tableau Server. You might want to preserve licensing if you are going to reinstall Tableau Server on the computer.

To completely remove Tableau Server without removing server licensing:

1. On the initial node, open a terminal session.

2. Run the tableau-server-obliterate script:

   sudo /opt/tableau/tableau_server/packages/scripts.<version_code>/tableau-server-obliterate -y -y -y

   **Note:** If you have already uninstalled Tableau Server and now you want to remove it, run the copy of the tableau-server-obliterate script located in the /var/tmp directory. If the script is not in that directory, reinstall the Tableau Server package and then run the tableau-server-obliterate script.

   If you have a multi-node (distributed) installation of Tableau Server, run the tableau-server-obliterate script on each node in the cluster.

3. Restart each computer you ran the tableau-server-obliterate script on.
To completely remove Tableau Server and licensing:

1. On the initial node, open a terminal session.

2. Deactivate any active product keys.

   tsm licenses deactivate -k <product_key>

3. Run the `tableau-server-oblate` script:

   sudo /opt/tableau/tableau_server-
   /packages/scripts.<version_code>/tableau-server-obliterate
   -y -y -y -l

   The `-l` option removes all licensing files from the computer. The script first attempts to deactivate any active licenses, but it will remove all licensing information whether or not deactivation was successful. This is why we recommend you run the `tsm licenses deactivate` command before running the obliterate script.

   **Note:** If you have already uninstalled Tableau Server and now you want to remove it, run the copy of the `tableau-server-obliterate` script located in the `/var/tmp` directory. If the script is not in that directory, reinstall the Tableau Server package and then run the `tableau-server-obliterate` script.

   If you have a multi-node (distributed) installation of Tableau Server, run the `tableau-server-obliterate` script on each node in the cluster. You do not need to deactivate licenses on any additional nodes.

4. Restart each computer you ran the `tableau-server-obliterate` script on.
Manage Sites

In addition to planning your sites in Tableau, you can manage users and groups, manage projects and control content access, manage your site data, and interact with views on the web.

What is a site ................................................................. 437
Planning a Site ............................................................. 440
Manage Users and Groups ............................................. 443
Dashboard-based Custom Portals .................................. 502
Manage Content Access ................................................ 506

What is a site

You might be used to using the term site to mean “a collection of connected computers,” or perhaps as the short form of “website.” In Tableau-speak, we use site to mean a collection of users, groups, and content (workbooks, data sources) that’s walled off from any other groups and content on the same instance of Tableau Server. Another way to say this is that Tableau Server supports multi-tenancy by allowing server administrators to create sites on the server for multiple sets of users and content.

All server content is published, accessed, and managed on a per-site basis. Each site has its own URL and its own set of users (although each server user can be added to multiple sites). Each site’s content (projects, workbooks, and data sources) is completely segregated from content on other sites.

If you are a server administrator on your Tableau Server deployment, you can learn more about sites, when to use them (vs. projects), and more in Sites Overview, in the Manage Server section.
Note: This article pertains to configuring sites on Tableau Server deployments. For Tableau Online, see Site Administrator Role and Tasks.

Site administrator tasks

Where the Server Administrator site role gives a user unrestricted access to the entire Tableau Server deployment, the Site Administrator site roles give a user unrestricted or minimally restricted access at the site level. The differences between Site Administrator Creator and Site Administrator Explorer are in the level of data connection and publishing access. Both site roles allow administering the site itself and managing site users. For more information, see Set Users’ Site Roles.

Although a server administrator can work at both the server and site levels, we make a distinction between the two levels of task. The site administrator is typically in charge of creating and maintaining the framework that enables Tableau users in the organization to publish, share, manage, and connect to data sources and workbooks. In this vein, site administrator tasks include any of the following (and both site roles allow this level of access):

- Creating project hierarchies to organize the site’s data sources and workbooks.
  This can include delegating project-level management to project leaders.

- Creating groups and assigning permissions that allow users to access only the content they need.

- Adding and removing users, assigning their site roles.
  This is allowed by default on a site; however, a server administrator can restrict this access to the server level only.

- Managing the site’s extract and subscription schedules.

- Monitoring site activity.
For more information about the distinction between server administrator and site administrator, see Administrator-level access to sites, in the Manage Server section.

Steps for setting up your site

The table below shows a loose sequence of steps for setting up a site, along with links to topics where you can get more information. You can complete the steps in any order that makes sense for you.

However, before you perform the steps to configure the site, we recommend spending some time with the articles in this section, learning about site authentication, site roles, projects, and permissions. Ideally you would document a plan for your projects, groups, and overall permissions strategy. Then set up a few projects and add a preliminary set of users, to test the plan and resolve issues before you add the remaining users. You can change many site settings after your users are working with the site, but try to go in with the intention of minimizing post-production changes.

<table>
<thead>
<tr>
<th>Plan</th>
<th>To supplement the recommendations above this table, get an overview of how the site components work together in Planning a Site.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Configure access</td>
<td>Work with the server administrator to determine how users sign in to the site, and configure the site appropriately.</td>
</tr>
<tr>
<td></td>
<td>For example, if the server is configured for single sign-on using SAML, you might configure SAML authentication at the site level as well.</td>
</tr>
<tr>
<td>Create projects and the permissions structure</td>
<td>Projects help you organize content, delegate project-level content management, and manage permissions effectively. To get started, see Use Projects to Manage Content Access.</td>
</tr>
</tbody>
</table>
### Planning a Site

Before you add users and content to a site, we recommend that you plan the following aspects of the site.

- Projects
- Users and groups

### Add users

Determine the users who can sign in to the site. See Add Users to a Site.

### Get your data to Tableau Server

After you create your projects and permissions structure, designate approved users for publishing and managing vetted data sources to the appropriate projects on the site.

In some organizations, people serve in multiple Tableau roles. Site administrators commonly also are data stewards. By that, we mean they create, publish, and manage the Tableau data connections. If this is you, make sure you are assigned the Site Administrator Creator site role.

After content is published to the site, you can maintain connection information (credentials, access tokens) and refresh schedules. For more information, see Refresh Data on a Schedule.

### Analyze site usage and performance

Monitor usage, performance, and other metrics. See Administrative Views.

---

Tableau Server on Linux Administrator Guide
The subsequent sections go over these site components, assuming that you are familiar with

**Note:** This article and section apply only to self-managed Tableau Server deployments on-premises or in the cloud. If you use Tableau Online, see [Manage Content Access](#).

## Projects

You can create projects on a site, which act as containers in which you can organize related content assets (such as data sources and workbooks). For example, you might set up a project to contain all of the certified data sources and workbooks your organization uses for mission-critical decisions. Or you might set up projects by department.

Projects are also useful for managing permissions. Once you know how your users need to access content, it’s usually easier to create projects based on those the type of content, and maintain permissions at the project level.

Every site has a default project named **Default**. When you create projects, the new projects get their initial set of permissions from the default project. In effect, the default project is a template for new projects. As we explain in related articles, for most environments, we recommend that you use the Default project only as a permissions template, and not as a container for published content.

For more information, see [Use Projects to Manage Content Access](#).

## Users and groups

Any user who will publish content or access published content on a site must be able to sign in to the site. If the user already has an account on the server, you’ll need to add that user to the appropriate site. You can add a user to more than one site as well. If the user doesn’t
already exist, you need to create a user account. Either way, make a list of the users who will need to be able to sign in to each site.

**Note:** The server license might restrict how many users you can have, or what level of access they can have. Check with the server administrator to make sure that you’ll be able to have an account for all your users.

In general, we recommend that you create groups on the server and then add users to the groups. This helps to make permissions much easier to manage. You can assign permissions on groups, to give those permissions to all users in the group. (See the next section.)

A typical strategy is to create groups for users who use content in similar ways. For example, you might create a group named SalesWBPublishers for all the users in the Sales department who publish workbooks, and a separate group named SalesDSPublishers for people in Sales who publish data sources. Each of these sets of users needs its own set of capabilities, so it makes sense to have a group for each for these needs.

### Site roles and permissions

Each user has a *site role* that determines the maximum permissions that they can have on the site. As part of your site planning, you need to decide each user’s site role. A user with a site role that’s too restrictive might not be able to do the work they need. By the same token, a security best practice is to limit users’ capabilities to only those that they need to do their work. This is referred to as following the principle of *least privilege*.

You or a site administrator you delegate this task to must also determine the permissions a user needs to work with content. Each content asset (workbook, data source, project) supports a set of *capabilities*. For example, you can View or Add Comments to a workbook. Before a user can perform tasks on a workbook, their permissions must allow those capabilities. A recommended practice is to sketch out a mapping of permissions to users outside of Tableau before you try to set this up on the server.
Permissions determine what a user can do within the context of the site role. A user whose site role is **Viewer** can never publish to the site, regardless of the permissions you grant them. A user whose site role is **Creator** can publish a workbook to the site, but only if that user has permission to save and view workbooks.

**Extract refresh schedules**

If users publish data sources or workbooks that include extracts, you usually want to make sure that the extracts are refreshed, so that they contain the latest data. Users can manually refresh an extract, but this isn’t always a good idea if the extract is large, and the refresh takes a long time. Instead, you can set up schedules for when an extract should be refreshed. Another planning task for a site administrator, therefore, is to think about when extracts should be refreshed and to work out schedules.

**Manage Users and Groups**

You can add users to your Tableau sites and set their site roles, which determines each user’s level of access. In addition, you can create groups of users, and enable guest access to your sites.

**Add Users to a Site**

Everyone who needs to access Tableau Server—whether to browse, publish, edit content or administer the site—must be added as a user. Administrators have the following options for adding users:

- Add a local user account or a user account from Active Directory, described later in this article.

  You can also add users by importing an Active Directory group. See Create Groups via Active Directory.
For Tableau Server on Linux, all external directory communication is configured and managed with a LDAP identity store. In the context of user and group synchronization, Tableau Server configured with LDAP identity store is equivalent to Active Directory. Active Directory synchronization features in Tableau Server function seamlessly with properly configured LDAP directory solutions.

- Import Users via a CSV file that you create using the CSV Import File Guidelines.

Site administrator access to user management

By default site administrators can add and remove users on a site. On the site’s Settings page, server administrators can revoke that capability, so that only server administrators can manage the site’s users.

A site administrator can edit an existing local user account only if the administrator has access to all of the sites the user is a member of. For example, if User1 is a member of sites A and B, an administrator of only site B cannot edit User1’s full name or reset the password.

Add local users to a site

1. Sign in to Tableau Server as an administrator, and if applicable select the site.

2. Select Users. On the Users page, click Add Users, and then click New User.
3. Enter a user name. With local authentication, using an email address for the user name is the best way to avoid user name collisions (for example, lrdriguez@example.com instead of lrodriguez).
User names are not case sensitive. Characters not allowed in user names include the semi-colon (;) and colon (:).

Also enter information in the following fields:

- **Display Name**—Type a display name for the user (e.g., Laura Rodriguez).
- **Password**—Type a password for the user.
- **Confirm password**—Retype the password.
- **Email**—This is optional and can be added at a later time in the user profile settings.

4. Select a site role.

   For site role definitions, see Set Users’ Site Roles.

5. Click **Add User**.
Add Active Directory users to a site

Before adding users to a site, be sure to review User Management in Active Directory Deployments to understand how multiple domains, domain naming, NetBIOS, and Active Directory user name format influence Tableau user management.

1. In a site, click Users, and then click Add Users, and then click Active Directory User.

2. Enter one or more user names (separated by semicolons).

   For example, tdavis; jjohnson; hwilson

   If you are adding a user that is from the same Active Directory domain that the server is running on, you can type the AD user name without the domain. The server's domain will be assumed.

   **Note:** Do not enter the user's full name in this field; it can cause errors during the importing process.

3. Select a site role.
For site role definitions, see Set Users’ Site Roles.

4. Click **Import Users**.

**Remove local users**

1. Sign in to Tableau Server as an administrator, select the site, and open the **Users** page.

2. Select the check boxes next to the users’ names, and on the **Actions** menu, select **Remove**.

   If a user is a member only of the current site, and they do not own any content, the user is removed from the server.

   If a user you remove from the current site is a member of other sites on the server, their status remains unchanged on those sites.

**Related information**

- You can also Add Users to Tableau Server, without site affiliation.

- Upgrading Tableau Server to version 2018.1 or later from a pre-2018.1 version, without activating user-based licenses, affects users who were assigned the **Viewer** site role in the pre-2018.1 server version.

  To learn more, see the section “User-based licenses” in the Licensing Overview and see Set Users’ Site Roles.

**Set Users’ Site Roles**

When you add users to a site on Tableau Server or Tableau Online, independent of their license type, you must apply a **site role** to them. The site role signifies the maximum level of access a user can have on the site. Along with content permissions, the site role determines who can publish, interact with, or only view published content, or who can manage the site’s users and administer the site itself.
How user licenses, site roles, and content permissions work together

The intersection of a user’s license type, site role, and content permissions determines the level of access a user has on the Tableau site.

1. The license type is associated with the user. The site role you want to assign to the user determines the license type they will require.

   In a multi-site environment on Tableau Server, a user’s license applies to all sites the user is a member of.

2. The site role is also set at the user level. In a multi-site environment, you assign site roles on each site. For example, the same user can have the Site Administrator Creator site role on one site, and Viewer site role on another site.

   The site role defines the maximum capabilities the user can have.

3. Whether the site role’s maximum capabilities are available to the user depends on the permissions set on the content resources (projects, data sources, workbooks).

For example, let’s say that a user has the following access on a site:

- Creator license (due to their access on another site)
- Explorer site role (on this site)
- Save permission capability on a project (on this site)

In this scenario, even though the license allows connecting to and creating new data sources in the web editing environment or Tableau Desktop, and a permission rule allows them to save in a project, their site role prevents them from being able to save so their effective permissions do not include the save capability. The user can’t publish content to the site.

Similarly, even if a user has a creator license and a creator site role, if they do not have the save capability on at least one project, they can’t publish anything to the site.
For more information, see Permissions.

Change a user’s site role

1. Sign in to the site as a server or site administrator, and go to the Users area.

   If you are a site administrator and do not see the Users area, check with your server administrator on whether they have denied user management capabilities to site administrators.

2. Select the users, and then select Actions > Site Role.

3. Select the new site role, and then click Change Site Role.
You can hover the pointer over the information icon to display a matrix that shows the maximum level of general capabilities each site role allows. For more information, continue to General capabilities allowed with each site role.

General capabilities allowed with each site role

The following table lists the license types as of version 2018.1, the highest level of site role allowed with each, how each site role maps to its pre-2018.1 equivalent; and summarizes the maximum capabilities each site role allows.

What this article covers and where to find what’s not covered here

- This information focuses on site roles and is more generalized. For a list of common specific tasks available per license role, see the matrix on the For Teams & Organizations tab on the Tableau pricing page.

- This information describes site roles as of version 2018.1. To learn more about how core-based licensing relates to user-based licensing, how licenses transfer, or other specific licensing transition scenarios, start with the following topics:

  Migrate from Core-Based to User-Based Licensing
  
  Troubleshoot Licensing
## Tableau site roles as of version 2018.1

<table>
<thead>
<tr>
<th>Site role name as of version 2018.1</th>
<th>Previous site role name</th>
<th>Maximum capabilities this site role allows</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Site roles that use a Creator license</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>— Users with these site roles have access to Tableau clients such as Tableau Prep, Tableau Desktop, Tableau Bridge, and Tableau Mobile.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Server Administrator</td>
<td>Server Administrator</td>
<td>Available on Tableau Server only; not applicable to Tableau Online.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This site role always occupies the highest license activated on the server between Creator and Explorer. It allows unrestricted access to the configuration settings for the Tableau Server browser environment, all sites on the server, users and groups, and all content assets, such as flows, projects, data sources (including connection information), and workbooks.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Connect to Tableau published data sources or external data, from the browser, Tableau Desktop, or Tableau Prep; create and publish new data sources; author and publish workbooks.</td>
</tr>
<tr>
<td>Site Administrator Creator</td>
<td>--</td>
<td>This is the highest level of access for Tableau Online.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Unrestricted access to content as described above, but at the site level. Connect to Tableau or external data in the browser, Tableau Desktop, or Tableau Prep; create new data sources; build and...</td>
</tr>
<tr>
<td>Site role name as of version 2018.1</td>
<td>Previous site role name</td>
<td>Maximum capabilities this site role allows</td>
</tr>
<tr>
<td>-----------------------------------</td>
<td>-------------------------</td>
<td>------------------------------------------</td>
</tr>
<tr>
<td>Creator</td>
<td>--</td>
<td>publish content.</td>
</tr>
</tbody>
</table>

On Tableau Server, server administrators can determine whether or not to allow site administrators to manage users and assign site roles and site membership. By default, on Tableau Server, and always on Tableau Online, site administrators are allowed these capabilities.

Creator: This is similar to the former Publisher site role, but allows new features. This site role offers non-administrators the maximum level of *content* access.

Connect to Tableau or external data in the browser, build and publish flows, data sources and workbooks, have access to Dashboard Starters, and use interaction features on published views. Can also connect to data from Tableau Prep or Tableau Desktop, publish (upload/save) and download flows, workbooks and data sources.

### Site roles that use an Explorer license

— Users with these site roles can access the server from the browser or Tableau Mobile.

<table>
<thead>
<tr>
<th>Server Administrator</th>
<th>N/A</th>
<th>Tableau Server only; not applicable to Tableau Online.</th>
</tr>
</thead>
</table>

If Explorer is the highest license type activated on the server when a new server administrator user is created, the user’s site role is Server Administrator;
### Tableau Server on Linux Administrator Guide

<table>
<thead>
<tr>
<th>Site role name as of version 2018.1</th>
<th>Previous site role name</th>
<th>Maximum capabilities this site role allows</th>
</tr>
</thead>
<tbody>
<tr>
<td>Site Administrator</td>
<td>Site Administrator</td>
<td>Same access to site and user configuration as Site Administrator Creator, but can’t connect to external data from the web editing environment. Can connect to Tableau published data sources to create new workbooks, and edit and save existing workbooks.</td>
</tr>
<tr>
<td>Explorer</td>
<td>墙面 Administrator</td>
<td>Can publish workbooks from the web using existing data sources, browse and interact with published views, use all interaction features, and can save new standalone data sources from data con-</td>
</tr>
</tbody>
</table>

however, the user will not have the full connecting and publishing capabilities that come only with the Creator license.

With the Explorer license a Server Administrator has unrestricted access to the configuration settings for the Tableau Server browser environment, all sites on the server, users and groups, and all content assets, such as projects, flows, data sources (including connection information), and workbooks.

However, with the Explorer license, a Server Administrator can’t connect to external data from the browser to create a new data source. They can author or publish workbooks and data sources from Tableau Desktop. (they function as an Explorer (can publish) site role with regards to publishing).
<table>
<thead>
<tr>
<th>Site role name as of version 2018.1</th>
<th>Previous site role name</th>
<th>Maximum capabilities this site role allows</th>
</tr>
</thead>
<tbody>
<tr>
<td>Explorer</td>
<td>Interactor</td>
<td>Can browse and interact with published views. Can subscribe to content, create data driven alerts, connect to Tableau published data sources and open workbooks in the web authoring environment for ad-hoc queries, but they can’t save their work.</td>
</tr>
<tr>
<td>Read Only</td>
<td>Viewer</td>
<td>This site role is available only in version 2018.1, for transitioning users to the user-based Viewer (or other) license and site role. Any users in the Read Only site role prior to upgrading to version 2018.2 or later are reassigned to the Viewer site role. In 2018.1 versions, Read Only users can see and subscribe to published views others have created. Can’t use other interaction features or save custom views.</td>
</tr>
</tbody>
</table>

**Site roles that use a Viewer license**

<table>
<thead>
<tr>
<th>Viewer</th>
<th>N/A</th>
</tr>
</thead>
<tbody>
<tr>
<td>Can see published views others have created and use most interaction features. Can subscribe to views and download as images or summary data. Can’t connect to data, create, edit, or publish con-</td>
<td></td>
</tr>
</tbody>
</table>
Site role name as of version 2018.1 | Previous site role name | Maximum capabilities this site role allows
--- | --- | ---

| Unlicensed | Unlicensed | Unlicensed users can’t sign in to Tableau Server or Tableau Online. Users are assigned the Unlicensed role in the following circumstances:

- You import users from a CSV file and their license level is set to unlicensed.
- The number of available licenses is reached at the time you add or import users.
- You remove a user who owns content on the site. The user will still own the content but not be able to do anything with it.

Other site roles

Who can publish content

The following site roles allow the specified level of publishing access.

- **Server Administrator** (Tableau Server only); **Site Administrator Creator**; and **Creator** allow full connecting and publishing access.
This includes connecting to data and publishing new flows, new workbooks and new data sources from Tableau Desktop and the web editing environment. The site roles also allow editing and saving existing published workbooks, or publishing updates to existing data sources.

- **Explorer (Can Publish)** and **Site Administrator Explorer** have limited publishing capabilities, as described in General capabilities allowed with each site role.

- **Explorer, Viewer, Read Only**, and **Unlicensed** do not allow publishing.

Site roles and Active Directory import and synchronization

When you import Active Directory users to a site, you can specify the site role. If a user is not yet a member of any site on the server, the user is added to the site with the assigned role. When you synchronize Active Directory groups, the site role is applied through the **Minimum Site Role** setting on the **Groups - Details** page.

**Note:** If you run Tableau Server on Linux, all external directory communication is configured and managed with a LDAP identity store. In the context of user and group synchronization, Tableau Server configured with LDAP identity store is equivalent to Active Directory. Active Directory synchronization features in Tableau Server function seamlessly with properly configured LDAP directory solutions.

If a user already exists in a Tableau Server site, the site role assigned during the import or sync process will be applied if it gives the user more access in a site. Importing or synchronizing AD users and groups can promote a user’s site role, but does not demote a user’s site role.

If a user already has the ability to publish, that ability is maintained.

The matrix below shows the rules applied for site roles on import.
**Note:** The **Import Site Role** row abbreviated headers indicate the site role specified for import. The **Current Site Role** column headers represent the current user site role. The table values represent the abbreviated resulting site role.

- Site Administrator: SA
- Site Administrator Creator: SC
- Site Administrator Explorer: SE
- Creator: C
- Explorer: E
- Explorer (Can Publish): EP
- Viewer: V
- Unlicensed: U

<table>
<thead>
<tr>
<th>Import Site Role</th>
<th>Current Site Role</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Site Administrator Creator</strong> (SC)</td>
<td>SC  C  SE  EP  E  V  U</td>
</tr>
<tr>
<td>SC</td>
<td>SC  SC  SC  SC  SC  SC  SC</td>
</tr>
<tr>
<td>Site Administrator Explorer (SE)</td>
<td>SE  SE  SE  SE  SE  SE  SE</td>
</tr>
<tr>
<td>Creator (C)</td>
<td>SC  C  SE  C  C  C  C</td>
</tr>
<tr>
<td>Explorer</td>
<td>SE  C  SE  EP  E  E  E</td>
</tr>
</tbody>
</table>
View, Manage, or Remove Users

Administrators can manage a site’s users such as adding and removing users, setting the groups they’re members of, setting their site roles, and so on. On Tableau Server, server administrators can manage users on multiple sites at a time on the All Sites page.

View and manage users on a site

Sign in to a site as an administrator, and then select Users. On this page you can do any of the following to manage users:

- Set group membership, set site role, or remove the user from the site. If you’ve configured the site for SAML single sign-on, you can set the selected users’ authentication type.
Select a user name to see details about them, such as content they own, views they subscribe to, and their account settings.

The user Settings page is available when the following conditions are true:

- The user is a member only of sites that the site administrator controls

- Site administrators can manage users. This is always the case for Tableau Online and is the default for Tableau Server; Tableau Server administrators can change this access for site admins.
If the server is configured to use the internal user management system (Local Authentication), you can edit the **Display Name**, **Email**, and **Password** for users after they have been added. If you are making many changes, you may find it easier to import the changes from a CSV file. For details, see Import Users and CSV Import File Guidelines.

**View and edit server users**

Sign in to Tableau Server as a server administrator. On the site menu, select **Manage All Sites**, and then select **Users**.
Manage users’ site membership

By default, server and site administrators can manage users at the individual site level. Server administrators can also manage users and their site roles on multiple sites. You do this at the All Sites level (at the server level).

1. In the site menu, select Manage All Sites, and then select Users.
2. On the Server Users page, select the check boxes next to the users, and then select Actions > Site Membership.
3. Select one or more sites, and a site role for each site, and then click Save.

Search for users

To search for a specific user

Use the filter toggle in the upper right to display the search box and site role filter. Then use the search box or filters to find the users you want.
The search operation checks the display name and user name attributes.

You can use the asterisk (*) character as a search wildcard. For example, searching for John* will return all user names that start with John.

Remove users from a site

You can remove a user only if the user does not own any content (projects, workbooks, views, or data sources). If you attempt to remove a user who owns content, the user site role will be set to Unlicensed, but the user will not removed.

**Note:** On Tableau Server, when an administrator removes a user from a site (and the user belongs only to that one site), the user is also deleted from the server.

1. Sign in to a site as an administrator, and go to the Users area. Select one or more users to remove, and then select Actions > Remove.
2. Click **Remove** in the confirmation dialog.

**Remove users from the server**

You can remove a user only if the user does not own any content (projects, workbooks, views, or data sources). If you attempt to remove a user who owns content, the user site role will be set to Unlicensed, but the user will not removed.

If a user is a member of multiple sites, and they own content on any of those sites, they are removed from the sites on which they don’t own content. The user remains a member on sites where they own content, but demoted to the Unlicensed site role.

1. In the site menu, click **Manage All Sites**, and then click **Users**. In a single-site environment, click **Users**.

   Select one or more users to delete, and then click **Actions > Delete**.
2. Click **Delete** in the confirmation dialog box.

**Change passwords for users of a single site**

To change the password for a user with membership to a single site, sign in to Tableau Server as a site administrator or a server administrator.

1. Ensure that the correct site is selected in the menu.

2. Click **Users**.

3. Click the display name of a user.

4. Click **Settings**.
5. Click the Change Password link, edit the password, and then click Save Password.

Change passwords for users of multiple sites

To change the password of a user with membership to multiple sites, sign in to Tableau Server as a server administrator.

1. In the site menu, click Manage All Sites.

2. Click Users.

3. Click the display name of a user.

4. Click the Change Password link, edit the password, and then click Save.
Set the User Authentication Type for SAML

On a site that has been configured for site-specific SAML, administrators can specify users’ authentication type. For example, if Tableau Server was configured for site-specific SAML and server-wide SAML, administrators can specify which users authenticate with site-specific SAML and which users authenticate with server-wide SAML.

You can assign authentication type at the time you add users to Tableau Server, as well as any time afterward.

1. When you’re signed in to the Tableau Server site, select Users.

2. On the Site Users page, select the check boxes next to the users you want to assign an authentication type.

3. On the Actions menu, select Authentication.
4. In the Authentication dialog box, select **Site SAML** or **Server Default**.

**Notes**

- Users that authenticate with site-specific SAML can only belong to one site. If a user needs to belong to multiple sites, set their authentication type to the server default. Depending on how site-specific SAML was configured by the server administrator, the server default is either local authentication or server-wide SAML.

- If you change users’ authentication to site-specific SAML, the next time they sign in, they will be directed to your identity provider’s site to provide their credentials.

**Import Users**

To automate the process of adding users to a site, you can create a CSV file that contains user information, and then import the file.

Site administrators can import users to a particular site; server administrators (Tableau Server only) can import users at the server level, to later add them to multiple sites.

**Note:** This topic contains the steps for importing, assuming that you have already created the CSV file. If you have not created the file yet, see CSV Import File Guidelines for a list of file format requirements and import options.
Add users from a CSV file

The following steps describe how to add users to a site or to the server. The images reflect adding users at the site level.

1. Do one of the following:
   - To add users at the site level, select Users, and then Add Users.
   - To add users at the server level on a single-site server, select Users, and then Add Users.
   - To add users at the server level on a multi-site server, open the list of sites, and select Manage All Sites. Select Users, and then Add Users.

2. Click Import From File, click Browse and navigate to the file, and then click Import Users.
Tableau displays the results of the import process (names in this image are blurred).

3. Click **Done**.

**How users’ site roles are assigned or maintained**

When you import at the site level or on a single-site server using `tabcmd`, you can specify the site role for all users in the CSV file. If a user already exists in the Tableau Server site, the site role assigned during the import process will be applied, even if it is more restrictive than users’ existing site role. The exception is that you cannot affect a server administrator’s site role.

For a multi-site server, when you import users in the Server Users page, you create server users with no site affiliation. Because these users do not belong to a site, they cannot have a site role. The only site role a user can have at the server level is **Unlicensed** or **Server Administrator**.

You can also specify the user’s site role when you assign site membership to a user. For information, see Manage users’ site membership.
Importing at the server level in multi-site environments

If the server is running multiple sites and you are a server administrator, you can import a CSV file from two locations. Where existing user accounts are concerned, each location has different capabilities.

- The **Server Users** page appears in a multi-site environment. Only server administrators can access this page.

![Server Users Page](image)

You can import the CSV file from here if you want to update existing user accounts in addition to adding new ones. For example, if you import a file that has a new password for each existing user, their passwords will be reset.

- The **Site Users** page.

![Site Users Page](image)

Server administrators can add new user accounts with CSV imports. If the CSV file includes existing users, the **Password** and **Display Name** fields must either match
the existing or be left blank. If new passwords or full names are used, the import will fail.

Importing to a single-site environment

Server and site administrators on a single-site server perform CSV user imports from the **Users** page in a site.

Multi-site versus single-site import

Users can belong to more than one site on the same server, but they must use the same credentials for each site. This becomes important when you’re adding users to a site and those users might already be members of a different site. If you try to import a user who already exists, and if the user’s credentials in the CSV file don’t match the existing credentials, the import fails for that user.

If you’re importing users to a site and think the users might already exist on the server, you can try leaving the **Password** field in the CSV file blank (while including the delimiters for the field). If a user who is defined in the CSV already exists in another site, the user is added to the site where you’re importing. However, if the user *doesn’t* already exist on the server, the user is created, and the CSV import window alerts you that the new user doesn’t have a password. You can then use the server administrator pages to assign a password to any user who doesn’t have one.
Note: If the server is configured to use Active Directory for authentication, user passwords are managed by Active Directory, and you can leave the password field in the CSV file blank.

CSV Import File Guidelines

You can automate adding users by creating a comma-separated values (CSV) file with user information and then importing the file. You can include attributes in the CSV file, such as license level and the publishing access, to apply to the users at the same time you import them.

To import users, you can use the server or site administration pages or the tabcmd utility. Using tabcmd provides an option for assigning a site role to all users in the CSV file. For information, see Import Users or createsiteusers filename.csv.

You can import users at the site or server level. If you import users to the server (not to a specific site), the users aren’t assigned to a site and are imported as Unlicensed.

CSV file format requirements

When you create the CSV file for importing users, make sure that the file meets the following formatting requirements:

- The file does not include column headings. Tableau Server assumes that every line in the file represents a user.

- The file is in UTF-8 format, and includes the byte-order mark (BOM).

- Character encodings such as BIG-5 have been converted to UTF-8. You can do this by opening the file in a text editor and using the Save As command.

- If a user name includes an @ character that represents anything other than a domain separator, you need to refer to the symbol using the hexadecimal format: \0x40
For example, user@fremont@mycompany.com should be user-
\0x40fremont@mycompany.com

Required columns in the CSV file

The following values are required for each user:

- User name

- Password: If Tableau Server is configured to use Active Directory authentication, there must be a Password column, but the column itself should be empty. If the server is using local authentication, you must provide passwords for new users.

Additional import file options

The CSV file can contain the following fields, in the order shown here:

- User name. The user name. If the server is configured to use Active Directory, this value must match a user defined in Active Directory. If the user name is not unique across domains, you must include the domain as part of the user name (for example, example\Adam or adam@example). This is the only required field.

- Password. A password for the user. If the server is configured to use Active Directory, this value is not used.

- Display name. The display name is part of the information used to identify a user on the server. If the user's display name is already in use, Tableau Server updates the existing user information with the settings in the CSV file. If the server is configured using Active Directory, this value is not used.

- License level. This can be Creator, Explorer, Viewer, or Unlicensed.

- Administrator level (System, Site, or None). This setting determines whether the user is imported as an administrator.
If you are using the web UI to import users, you can set the administrator site role to **System** only if you import the file at the server (All Sites) level. If you are signed in to a specific site, and if the administrator column for a user in the CSV file is set to **System**, Tableau Server imports the user as a site administrator.

- Publishing capability (**yes/true/1** or **no/false/0**). If you are using the web UI, the publisher setting is used only if you import while signed in to a specific site. If you import users at the server (All Sites) level, this value isn’t used.

- Email address. The email address is part of the information used to identify a user on the server. If the email address is already in use, Tableau Server updates the existing user information with the settings in the CSV file.

The order of the columns is significant. The first column is treated as the user name, the second as the password, the third as display name, and so on, regardless of the content in the columns. If you omit values for a field, you must still include the field’s comma delimiter.

**Improve performance for large CSV files passed through tabcmd**

A server administrator can enable server settings that help to improve performance for importing large CSV files through tabcmd commands. You can do this using the `tsm configuraiton set` command with the following options:

- `vizportal.csv_user_mgmt.index_site_users`
- `vizportal.csv_user_mgmt.bulk_index_users`
- `searchserver.index.bulk_query_user_groups`

Essentially, these options index users after the CSV file is processed, instead of one-by-one as they are added to the server’s database. This reduces the number of calls to the database and the memory required to process the file. These `tsm configuration set` options apply to the `tabcmd createsiteusers,deletesiteusers,addusers, and removeusers` commands.

For descriptions for these settings, see `tsm configuration set Options`. 
Notes

- If you are not signed in to a specific site and are importing users at the server level, you can assign only the Server Administrator and Unlicensed site roles.

- If you have a user-based server installation, and if adding users would exceed the number of users allowed by your license, the users are added as unlicensed users.

- If you use `tabcmd` and specify the license, but importing users would exceed your license limits, users are imported as Unlicensed.

CSV settings and site roles

The license level, administrator, and publishing settings for a user determine how the user’s site role is set during the import process. The following table shows how the settings are converted to site roles.

<table>
<thead>
<tr>
<th>CSV settings</th>
<th>Site role</th>
</tr>
</thead>
</table>
| License level=(any)            | Server Administrator. This setting applies to Tableau Server only, and it is valid only if you are importing users while managing the server (that is, not signed in to a specific site). The Server Administrator site role always takes a 
  Creator license if one is available. If no Creator license is available, see Troubleshoot Licensing to learn about the way Tableau Server handles this. |
| Administrator=System Publisher=true | Site Administrator Creator or Site Administrator Explorer. This setting is valid only if you are importing users while signed in to a specific site. |
| License level=Creator or Explorer  Administrator=Site Publisher=true | Creator |
### CSV settings vs. Site role

<table>
<thead>
<tr>
<th>CSV settings</th>
<th>Site role</th>
</tr>
</thead>
<tbody>
<tr>
<td>Administrator=None</td>
<td></td>
</tr>
<tr>
<td>Publisher=true</td>
<td></td>
</tr>
<tr>
<td>License level=Explorer</td>
<td>Explorer (Can Publish)</td>
</tr>
<tr>
<td>Administrator=None</td>
<td></td>
</tr>
<tr>
<td>Publisher=true</td>
<td></td>
</tr>
<tr>
<td>License level=Explorer</td>
<td>Explorer</td>
</tr>
<tr>
<td>Administrator=None</td>
<td></td>
</tr>
<tr>
<td>Publisher=false</td>
<td></td>
</tr>
<tr>
<td>License level=Viewer</td>
<td>Viewer</td>
</tr>
<tr>
<td>Administrator=None</td>
<td></td>
</tr>
<tr>
<td>Publisher=false</td>
<td></td>
</tr>
<tr>
<td>License level=Unlicensed</td>
<td>Unlicensed</td>
</tr>
<tr>
<td>Administrator=None</td>
<td></td>
</tr>
<tr>
<td>Publisher=false</td>
<td></td>
</tr>
</tbody>
</table>

### CSV import example for Tableau Server

The following example shows a CSV file that contains information for several users.

```
henryw,henrypassword,Henry Wilson,Creator,None,yes,henryw@example.com
freds,fredpassword,Fred Suzuki,Viewer,None,no,freds@example.com
alanw,alanpassword,Alan Wang,Explorer,Site,yes,alanw@example.com
```
michellek,michellepassword,Michelle Kim,Creator,System,yes,michellek@example.com

If you import this file while managing a site, four users are added to that site. The Administrator setting for user Michelle is System. However, because you are importing the users into a site, Tableau Server give Michelle the Site Administrator Creator site role. Three of the users are allowed to publish.

If you import this file while managing the server, four users are added to the server, but they are not added to any site. Only one user is imported as a server administrator; the rest are set to Unlicensed.

Manage Site User Visibility

By default, all site users can see aliases, project ownership and comments by other users when permissions allow. The User Visibility setting lets administrators manage if users with Viewer and Explorer site roles see other users and groups on the site, which can be important for sites that are used by multiple clients. To learn more about site roles, see Set Users’ Site Roles.

Limit user visibility

Setting User Visibility to Limited impacts certain collaboration tools and hides user information in Tableau Online and Tableau Server. Limited User Visibility either disables the feature for Viewers and Explorers (excluding Site Administrator Explorers), or removes user information from other areas. Note that Creators and administrators will still see user information when User Visibility is set to Limited.

To limit user visibility for Explorers and Viewers (excluding Site Administrator Explorers):

- Navigate to the site’s Settings page
- Select Limited in the User Visibility setting
The following is a list of site areas impacted when User Visibility is set to Limited. Unless noted that the feature is disabled for all users, only non-administrator Explorers or Viewers are impacted.

<table>
<thead>
<tr>
<th>Area</th>
<th>Impact</th>
</tr>
</thead>
<tbody>
<tr>
<td>Search</td>
<td>User information not displayed</td>
</tr>
<tr>
<td>Content owners</td>
<td>User information not displayed (Explorers and Viewers can't see themselves, but can see their content in My Content)</td>
</tr>
<tr>
<td>Profile pictures</td>
<td>User information not displayed</td>
</tr>
<tr>
<td>Subscriptions</td>
<td>User information not displayed</td>
</tr>
<tr>
<td>Recommendations</td>
<td>Similar users not displayed (all users)</td>
</tr>
<tr>
<td>Add/Edit Tags</td>
<td>Explorers and Viewers can see tags but cannot delete or modify them.</td>
</tr>
<tr>
<td>&quot;Who has seen this view?&quot;</td>
<td>Disabled</td>
</tr>
<tr>
<td>Ask Data usage analytics</td>
<td>Disabled</td>
</tr>
<tr>
<td>Permissions dialogs</td>
<td>Disabled</td>
</tr>
<tr>
<td>Named sharing</td>
<td>Disabled (all users)</td>
</tr>
<tr>
<td>Alerts</td>
<td>Disabled (all users)</td>
</tr>
<tr>
<td></td>
<td>Existing alerts are paused.</td>
</tr>
<tr>
<td>Comments</td>
<td>Disabled (all users)</td>
</tr>
<tr>
<td>Public Custom</td>
<td>Disabled (all users)</td>
</tr>
</tbody>
</table>
Views

Existing public custom views appear as private.

Request Access

Disabled (all users)

Tableau Desktop

Publishing workbooks disabled from Desktop.

User information not displayed on user filters.

Tableau Catalog (with Data Management Add-on)

User Visibility does not hide user information in certain Tableau Catalog features, such as Data Details, certification and contacts on external assets (databases and tables), lineage, data quality warnings, and related calls to the Tableau Metadata API and the Tableau Server REST API. For more information, see User visibility in the Tableau Metadata API.

**Note:** Tableau Catalog is enabled by default in Tableau Online with the Data Management Add-on.

External Asset contacts and certifications (with Data Management Add-on)

Tableau Server and Tableau Online sites with Tableau Catalog enabled: User information displayed

REST API and Metadata API with Data Management Add-on: User information displayed.

When User Visibility is set to Limited, Tableau Server REST API calls behave as described in the table above.

**Note:** If a user is a member of multiple sites, entering an email on the sign in page for Tableau Online will return the names of all sites the user is a member of.

Best practices for limiting user visibility

Administrators can also check that user and group information is not visible in these ways:
Configure permissions to only provide content to appropriate parties. For more information, see Permissions.

- Limited User Visibility hides user identification information from search, but might return content that the user published, including when searching by owner name, if the person searching has viewing permission to that content.
- A user publishing a workbook with a duplicate title in the same project might see a warning that a workbook with that title already exists.
- Apply row-level security when necessary.
- Check that metadata within dashboards does not contain user information.
- Check that calculations accessible to users don't contain user metadata (e.g., user filters).

User visibility in the Tableau Metadata API

In Tableau Online, Metadata API is always enabled. With the Data Management Add-on, user information can be exposed through contacts and certifications on external assets.

In Tableau Server, the Metadata API is disabled by default. When the Metadata API is enabled and User Visibility is set to Limited, Metadata API calls behave as described in the table above.

Restore Full User Visibility

When administrators set User Visibility back to Full, features disabled for all users by Limited User Visibility (such as comments and alerts) remain off. Administrators can re-enable these features through the site's Settings page.

Any previous feature settings are not retained when User Visibility is set to Full, and affected features are not automatically turned on.

Guest User

Core-based licenses of Tableau Server include a Guest user option, which you can use to let people access Tableau views without an account on the server.
Guest user access is enabled by default when Tableau Server is installed with a core-based license. It is not available with user-based licensing. If you do not intend to use Guest user access, you should disable it.

Guest access allows users only to see and interact with Tableau views. The Guest user cannot browse the Tableau Server interface or see server interface elements in the view, such as user name, account settings, comments, and so on. For more information about licenses, see Manage Licenses.

**Tip:** To share views with Guest users, either provide URL links or embed views into web pages. For more information, see Tableau User Help.

### Guest user permissions

A Guest user can have the following maximum capabilities:

- **Workbooks and views:** View, Export Image, Summary Data, View Comments, Filter, Full Data, Web Edit, Download (to save a local copy)

- **Data sources:** View and Download

If you add the Guest user to a group that has a higher level of access to a content resource, the Guest user’s access does not exceed the capabilities listed above. However, the Guest user account will comply with more restrictive permissions settings.

### Enable or disable Guest access

You must be a server administrator to change Guest account settings at either the server or the site level.

**Note:** Enabling the Guest user for a site can increase the number of potential simultaneous viewers beyond the user list you might be expecting. The administrative view Status > Traffic to Views can help you gauge the activity.
1. In the site menu, click Manage All Sites and then click Settings > General.

2. For Guest Access, select or clear Enable guest access.

3. Click Save.

This enables the Guest user on all sites. You can then go to the same setting for a specific site. To disallow Guest access for a site:

1. In the site menu, select a site.

2. Click Settings, and on the General tab, clear the Enable guest access for this site check box.

If the Guest account is enabled on some or all sites, and you turn off Guest access at the server level, it is turned off for all sites as well.

**Note:** You can turn off Guest user access at the server or site level; however, you aren’t able to remove the user. So, although no one can access data or views without signing in to the server, the Guest user still appears in the Site Users list and group lists for groups you’ve added the Guest user to.

**Additional Guest account characteristics**

The Guest user is unique in the following additional ways:

- As a single user account, it represents all unauthenticated users accessing Tableau views.

- When enabled, it is a member of the All Users group.

- You can add it as a member of other groups on a site.

- You cannot edit it or select it as the owner of a content resource.

- If the Guest user needs to access a workbook with an extract connection, the Guest
must also have the **View** capability on the published data source. (The Guest user is not allowed to connect to published data sources.)

- The account is not allowed to save custom views.
- Guest cannot be used in a user filter.
- You cannot delete the account; however, you can turn off access to it by clearing the check box described in the steps above.

**Groups**

You can create and delete user groups, add users to a group, and synchronize groups with Active Directory.

**Add Users to a Group**

You can organize Tableau Server users into groups to make it easier to manage multiple users. You can create groups on the server or import groups from Active Directory.

---

**Note:** If you run Tableau Server on Linux, all external directory communication is configured and managed with a LDAP identity store. In the context of user and group synchronization, Tableau Server configured with LDAP identity store is equivalent to Active Directory. Active Directory synchronization features in Tableau Server function seamlessly with properly configured LDAP directory solutions.

To keep Active Directory group membership up-to-date:

- Site administrators can synchronize selected groups on demand in a site. For more information, see *Synchronize Active Directory Groups on a Site*.

- Server administrators can synchronize all Active Directory groups on the server based on a schedule or on-demand. For more information, see *Synchronize All Active Directory Groups on the Server*. 

---
To add a user to a group, the group must already exist.

Add users to a group (Users page)

1. In a site, click **Users**.

2. Select the users you want to add to a group, and then click **Actions > Group Membership**.

3. Select the groups and then click **Save**.

Add users to a group (Groups page)

1. In a site, click **Groups**, and then click the name of the group.

2. In the group's page, click **Add Users**.
3. Select the users to be added, and then click **Add Users**.

**Create a Local Group**

Local groups are created using the Tableau Server internal user management system. After you create a group you can add and remove users.

1. In a site, click **Groups**, and then click **New Group**.

2. Type a name for the group and click **Create**.
Create Groups via Active Directory

You can import Active Directory groups to create matching groups on Tableau Server, as well as a user for each member of an Active Directory group that is not already on the server.

**Note:** If you run Tableau Server on Linux, all external directory communication is configured and managed with a LDAP identity store. In the context of user and group synchronization, Tableau Server configured with LDAP identity store is equivalent to Active Directory. Active Directory synchronization features in Tableau Server function seamlessly with properly configured LDAP directory solutions.

Each user is assigned a site role as part of the import process. If any of the users to be imported exist in Tableau Server, the site role assigned during the import process is applied only if it gives the user more access to the server. Importing users does not demote site roles.

Before importing groups, review User Management in Active Directory Deployments to understand how multiple domains, domain naming, NetBIOS, and Active Directory user name format influence Tableau user management.

1. In a site, click **Groups**, and then click **Add Groups**

2. Type the name of the Active Directory group you want to import, and then select the group name in the resulting list.
3. Select the site role for the users.
4. Click **Import**.

**Note:** You cannot change the name of groups imported from Active Directory. The group name can only be changed in Active Directory.

**Synchronize Active Directory Groups in a Site**

At any time, you can synchronize an Active Directory group with Tableau Server to ensure new users in Active Directory are also added in Tableau Server. You can synchronize individual groups or multiple groups at once.

**Note:** If you run Tableau Server on Linux, all external directory communication is configured and managed with a LDAP identity store. In the context of user and group
synchronization, Tableau Server configured with LDAP identity store is equivalent to Active Directory. Active Directory synchronization features in Tableau Server function seamlessly with properly configured LDAP directory solutions.

1. In a site, click **Groups**.
   
   On the Groups page, select one or more groups.

2. Click **Actions > Synchronize**.

![Groups page](image)

Set the minimum site role for users in an Active Directory group

In the **Groups - Details** page, administrators can set the minimum site role for group users to apply during synchronization.

This setting does not run synchronization; it sets the minimum site role to applied to the group every time synchronization runs. When you synchronize Active Directory groups, new users are added to the site with the minimum site role. If a user already exists, the minimum site role will be applied if it gives the user more access in a site. If you don’t set a minimum site role, new users are added as **Unlicensed** by default.

**Note:** A user’s site role can be promoted but never demoted based on the minimum site role setting. If a user already has the ability to publish, that ability will always be
Tableau Server on Linux Administrator Guide

maintained. For more information on minimum site role, see Site roles and Active Directory import and synchronization.

1. In a site, click Groups.

2. On the Groups page, select a group, and then select Actions > Minimum Site Role.

3. Select the minimum site role, and then click Change Site Role.

What happens when users are removed in the source Active Directory?

Users cannot be automatically removed from the Tableau Server through an Active Directory sync operation. Users that are disabled, deleted, or removed from groups in Active Directory remain on Tableau Server so that administrators can audit and reassign the user's content before removing the user's account completely. For more information, see Sync behavior when removing users from Active Directory.

What happens when a user name changes in the source Active Directory

By default, Tableau Server will not synchronize changes to the user display name or email address after the initial synchronization when the corresponding account is created in
Tableau Server. For example, if the user name jsmith is used for the display name John Smith, changing the display name in Active Directory to Joe Smith will not synchronize to the corresponding jsmith user in Tableau Server. Similarly, if the user's email changes in Active Directory, Tableau Server will not synchronize changes.

You can configure Tableau Server to update the name and email properties when they change in the source Active Directory by setting `vizportal.adsync.update_system_user` to true.

To change this behavior run the following tsm commands:

```
tsm configuration set -k vizportal.adsync.update_system_user -v true
tsm pending-changes apply
```

The `pending-changes apply` command displays a prompt to let you know this will restart Tableau Server if the server is running. The prompt displays even if the server is stopped, but in that case there is no restart. You can suppress the prompt using the `--ignore-prompt` option, but this does not change the restart behavior. For more information, see `tsm pending-changes apply`.

What happens when an Active Directory group is removed from Tableau Server?

Many Tableau administrators use Active Directory groups to import and create users. After the users are imported into Tableau Server, administrators will then delete the group in Tableau Server. Deleting a group does not delete the users in it.

**Synchronize All Active Directory Groups on the Server**

As a server administrator, you can synchronize all Active Directory groups (that have been configured on Tableau Server) on a regular schedule or on-demand on the **General** tab of the **Settings** page for the server.
**Note**: If you run Tableau Server on Linux, all external directory communication is configured and managed with a LDAP identity store. In the context of user and group synchronization, Tableau Server configured with LDAP identity store is equivalent to Active Directory. Active Directory synchronization features in Tableau Server function seamlessly with properly configured LDAP directory solutions.

Before you begin

Before synchronizing groups as described in this topic, you must first import the Active Directory group into Tableau Server. See Create Groups via Active Directory.

Synchronize Active Directory groups on a schedule

1. **Single-site**: Click **Settings** > **General**.

   **Multisite**: In the site menu, click **Manage All Sites** and then click **Settings** > **General**.

2. Scroll down the page to **Active Directory Synchronization**, and then select **Synchronize Active Directory groups on a regular schedule**.
3. Select the frequency and time of synchronization.

4. Click **Save**.

Synchronize all Active Directory groups on demand

At any time, you can synchronize Active Directory groups with Tableau Server to ensure that new users and changes in Active Directory are reflected in all Active Directory groups on Tableau Server.

1. **Single-site**: Click **Settings > General**.

   **Multisite**: In the site menu, click **Manage All Sites**, and then click **Settings > General**.
2. Under **Active Directory Synchronization**, click **Synchronize All Groups**.

View synchronization activity

You can view the results of synchronization jobs in the **Background Tasks for Non Extracts** administrative view. **Queue Active Directory Groups Sync** is the task that queues and indicates the number of **Sync Active Directory Group** tasks to be run.

1. **Single-site**: Click **Status**.

   **Multisite**: In the site menu, click **Manage All Sites** and then click **Status**.

2. Click the **Background Tasks for Non Extracts** link.

3. Set the **Task** filter to include **Queue Active Directory Groups Sync** and **Sync Active Directory Group**.

You can quickly navigate to this administrative view by clicking the **View synchronization activity** link in the **Settings** page for the server.

Set the minimum site role for users in an Active Directory group

In the **Groups - Details** page, you can set the minimum site role for group users to be applied during Active Directory synchronization.
This setting does not run synchronization; instead, it sets the minimum site role to applied to the group every time synchronization runs. The result is that when you synchronize Active Directory groups, new users are added to the site with the minimum site role. If a user already exists, the minimum site role is applied if it gives the user more access in a site. If you don’t set a minimum site role, new users are added as **Unlicensed** by default.

**Note:** A user’s site role can be promoted but never demoted based on the minimum site role setting. If a user already has the ability to publish, that ability will always be maintained. For more information on minimum site role, see Site roles and Active Directory import and synchronization.

1. In a site, click **Groups**.
2. On the Groups page, select a group.
   
   Click **Actions > Minimum Site Role**.
3. Select the minimum site role, and then click **Change Site Role**.
What happens when users are removed in the source Active Directory?

Users cannot be automatically removed from the Tableau Server through an Active Directory sync operation. Users that are disabled, deleted, or removed from groups in Active Directory remain on Tableau Server so that administrators can audit and reassign the user’s content before removing the user’s account completely. For more information, see Sync behavior when removing users from Active Directory.

Improving group synchronization performance

Active Directory synchronization is performed by the backgrounder process. The Backgrounder process is the same process that is used for managing and creating extracts, and is also used to generate subscription content. In large organizations with dynamic group membership and heavy extract usage, the Active Directory group synchronization process may be disruptive. We recommend running group synchronization during non-business hours.

By default, the Backgrounder process performs synchronization in a serial operation. This means that each group is synchronized, one after the other, in a single Backgrounder process. If you are running multiple instances of Backgrounder processes either on a single Tableau Server or across a distributed deployment, consider enabling parallel processing for Active Directory synchronization. When parallel Backgrounder processing is enabled, the group synchronization is distributed across multiple Backgrounder processes for better performance.

To enable parallel backgrounder processing for group synchronization, open TSM CLI and enter the following commands:

```
 tsm configuration set -k backgrounder.enable_parallel_adsync -v true
 tsm pending-changes apply
```

Quick Start: Synchronize All Active Directory Groups on a Schedule

After you import Active Directory groups in Tableau Server, you can make sure they stay synchronized in Tableau Server by setting up a schedule. You can also synchronize all Active
Directory groups on the server on-demand, at any time. The minimum site role setting for the group is applied when users are synchronized.

**Note:** If you run Tableau Server on Linux, all external directory communication is configured and managed with a LDAP identity store. In the context of user and group synchronization, Tableau Server configured with LDAP identity store is equivalent to Active Directory. Active Directory synchronization features in Tableau Server function seamlessly with properly configured LDAP directory solutions.

1 Set a minimum site role for synchronization

In a site, click **Groups**. Select a group, and then click **Actions** > **Minimum Site Role**. Select the minimum site role, and then click **Change Site Role**. Server and site administrators can set the minimum site role for group users to be applied during Active Directory synchronization. If you don't set a minimum site role, new users are added as **Unlicensed**.

Synchronizing can promote a user's site role, but will never demote a user's site role.
Set the schedule

Server administrators can enable synchronization for all Active Directory groups on the General tab of the Settings page for the server. Enable synchronization, select the frequency settings, and then click Save.

All Active Directory groups on the server are synchronized according to the same schedule.

Run synchronization on-demand (optional)

On the General tab of the Settings page, click Synchronize All Groups to synchronize all Active Directory groups on Tableau Server immediately. Click this button at any time to ensure new users and changes are reflected in all Active Directory groups on the server.
Click **Synchronize All Groups** to synchronize all Active Directory groups on the server outside of a schedule.

4 View the status of synchronization tasks

Server and site administrators can view the results of Active Directory synchronization jobs in the **Background Tasks for Non Extracts** administrative view. On the server or in a site, click **Status**. Under **Analysis**, click **Background Tasks for Non Extracts** and filter on the **Queue Active Directory Groups Sync** and **Sync Active Directory Group** tasks.
Queue Active Directory Groups Sync queues the Sync Active Directory Group tasks to be run.

Delete Groups

You can delete any group from the server (with the exception of the All Users group). When you delete a group, the users are removed from the group but they are not deleted from the server.

1. In a site, click Groups.

2. On the Groups page, select one or more groups to delete.
3. Select **Actions > Delete**.

### Dashboard-based Custom Portals

**Note:** This overview was inspired by the work of Tableau Zen Master Mark Jackson. For more details about the process, check out Mark’s blog.

The **standard Tableau Online or Tableau Server interface** works great for many organizations, but if you want to create a company-branded experience and don't have API developer skills, consider creating a custom portal based on a Tableau dashboard. A custom portal lets you organize content around specific departments or job roles, and you can even incorporate training that helps people interact with data more fully. As the volume of Tableau content grows, a custom portal guides your users directly to the data they need, while providing ready access to related views and Tableau Server search so they can easily explore further.
Sketch out a portal design

Begin outside of Tableau, on paper or in a wireframing application. Consider the structure of your organization and the number of Tableau views that apply to each group of users. Do you simply need one level of navigation that points immediately to content? Or might you need to start with a main page that provides navigational links for separate groups of users or report types, followed by a second level with related dashboards and views?

<table>
<thead>
<tr>
<th>Medical Staff</th>
<th>Illness trend report</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Patient demographics</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Facilities</th>
<th>Hospital occupancy</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Real estate transactions</td>
</tr>
</tbody>
</table>

After you get buy-in on your design from key stakeholders and data consumers, you’re ready to move on to the next phase.

Gather images for logos and navigation elements

Start thinking about images while refining your initial mockup, and then gather them from libraries of clip-art or approved brand graphics, or create them from scratch in an application like Photoshop or Snagit. PowerPoint can also be a good source if you plan to create navigation thumbnails of common chart types.
Lay out text, images, and selected sheets on a dashboard

Create a workbook with a dashboard for the portal, using a tiled layout for more predictable positioning and scaling of elements on different devices. Then start adding Text and Image objects, as well as any sheets for data views you want to highlight right up front in your portal. To refine spacing between these dashboard elements, insert Blank objects and adjust their size. For more information, see Create a Dashboard.

Tip: For dynamic elements that show popular or recently created content, include sheets based on custom administrative views.

Link dashboard elements to content

If you have a second level of navigation in your portal, use filter actions to point to a secondary dashboard from the main one. To create links that directly open data views, right-click Image objects and choose Set URL. (In our example below, each colored arrow and accompanying text is an image that links to a view URL.) You can even point to empty views with preloaded data sources, encouraging users to create new Tableau content in the web-authoring workspace.

Tip: To add hyperlinks to text objects, include the full URL (for example, http://www.tableau.com).
Publish, test, and refine your portal

Publish the workbook to your server and distribute the dashboard URL to your users. Portal design, like data analysis itself, is a cyclical process. Now that your portal is out in the wild, start gathering user feedback so you can continuously improve the experience.
Manage Content Access

You can manage who can access content on your site and set the permissions that govern content ownership.

Set a Site’s Web Authoring Access and Functions

Tableau Server administrators can specify at the site level whether to allow users to edit published views in the web environment and configure other web authoring functionality.

By default web authoring functionality is enabled for all sites. Users with the Web Edit capability can create and edit workbooks directly on the server. Turn off web authoring if you want users to be able to view and interact with published workbooks but not make any changes to the core information.

The steps below describe how to set web authoring and other associated functionality for an entire site. For more granular control over which users can use web editing, you can use projects, groups, and permissions. See Set Web Edit, Save, and Download Access on Content.

Turn web authoring on or off for a site

1. In a web browser, sign in to the server as an administrator and go to the site in which you want web authoring to be enabled. In that site, click Settings.

2. Select Allow users to use web authoring to enable the functionality.

   Clear the check box to turn off web authoring for that site.

Tip: As a finishing touch, hide the Tableau toolbar to give your portal a custom feel.

After the question mark at the end of the dashboard URL, add :embed-d=y&amp;toolbar=n
3. If your site is already in production, and you want the change to take effect immediately, restart the server.

Otherwise, the change takes effect after server session caching expires or the next time users sign in after signing out.

Notes

- When you enable web authoring, make sure that, on the appropriate workbooks or views, the permission rule for a user or group allows the Web Edit capability.

- If you turn off web authoring on a production site and do not complete the last step to restart the server, users might continue to have authoring access until their session caches expire or they sign out.

See which sites allow web authoring

To confirm which sites allow web authoring, on the site-selection menu at the top, select Manage All Sites, and then go to the Sites page.
Configure cross-database join options

To improve performance for cross-database joins, users can allow Tableau to perform the join using the live database they are connected to instead of using Hyper. While this option is faster, if Tableau uses the connected database to perform the join, data from the file data source that the user is connected to is temporarily moved into temp tables in the database. Because this moves data outside of Tableau, as an administrator you may want to restrict access to this feature for users with web authoring permissions.

1. In a web browser, sign in to the server as an administrator and go to the site in which you want web authoring to be enabled. In that site, click **Settings**.
2. In the **Cross-Database Joins** setting, select one of the following options:
   - **Use Tableau or existing databases** - Select this option if you want to allow users to choose whether they want to allow Tableau to use the live database to perform cross database joins. Published data sources with this option enabled will continue to use the user's database for cross-database joins.
   - **Use Tableau only** - Select this option to restrict users to use only Hyper to perform cross data-base joins.

   ![Cross-Database Joins](image)

If you select **Use Tableau only**, the option to choose how Tableau performs the cross-database join won't display in the canvas when the user connects to a supported data source and supported database. For more information about this feature, see **Improve performance for cross-database joins**.

![Cross-database join](image)
Set Web Edit, Save, and Download Access on Content

If you’re enabling web authoring functionality on your site, you can configure more precisely which users on the site have access to this functionality. Using site roles and permissions rules at the content level, you can grant or deny Web edit, Save, or Download capabilities on projects, workbooks, and data sources.

Note: This document strives to use the phrase Web edit only to specify the name of the capability in permissions rules, and web authoring to refer to the general functionality of creating and modifying workbooks on the server. However, you might otherwise see these two phrases used interchangably.

Why allow users to work on the site directly

As an administrator, your initial thought about allowing people to populate a site with content, seemingly indiscriminately, might be one of skepticism. However, with a few controls, you can limit where this is done, while providing important benefits that centralized content management offers both you and your users.

Web authoring pros and cons

For publishers and business users, some benefits of web authoring include the following:

- It provides analyst teams who work collaboratively with a central location in which to provide input.
- It enables people who do not have Tableau Desktop to connect to data sources and create workbooks.
- It enables people to access content when they are away from their Tableau Desktop computer or VPN, whether on a computer or a hand-held device.
- It can provide a framework for enabling consistency across Tableau reports. (By making template workbooks available on the site, analysts can download or create new workbooks with data connections, branding, and formatting already in place.

For administrators, benefits can include the following:
- Fewer Tableau Desktop deployments to manage and support.
- Fewer computers that need to have database drivers installed.
- Capacity to govern content.
- More accurate monitoring of what people are doing with Tableau.

Some disadvantages to web editing include the following:

- For analysts, web editing functionality is not as extensive as in Tableau Desktop (although it continues to evolve toward that parity).
- For administrators, more people working on the server might mean upgrading systems.
- Without publishing guidelines, content proliferation on the site is expected. This can confuse the people who rely on published Tableau dashboards and data sources, degrade server performance and data quality, and potentially affect data security.

Managing permissions to help users avoid content proliferation

To help users to avoid content proliferation on the site, many Tableau administrators use projects to allow varying levels of access to content. For example, one project can be configured to allow all users to edit and save workbooks; another can allow only approved publishers to save new content.

To get a better idea how this works, see the following resources:

- Configure Projects, Groups, and Permissions for Managed Self-Service
- Projects and Content Permissions in Everybody’s Install Guide
- Governed Self-Service at Scale, a Tableau whitepaper by Rupali Jain. To view the PDF, you might need to provide your Tableau website credentials. These are the same ones you use for the community forums or to submit support cases.

Coordinate edit and save capabilities with site roles for the appropriate level of access

To edit, save, and download workbooks, users must have a site role that allows those actions, along with the capabilities—defined in permissions rules—that grant or deny editing-related access.
Site role access

- When the appropriate permissions are set at the content level, the Creator or Explorer (can publish) site role allows both Save (overwrite) and Save As/Download.

Note that File > Save is only available to the workbook owner. When the Save permission capability has been granted at the project and workbook level, a non-owner user can overwrite the existing workbook in web authoring by selecting File > Save As and using the same workbook name. This overwrites the existing content and they become the owner and gain full access to the content.

- The Explorer site role can be granted the Web Edit and Save As/Download capabilities, but they will not be able to save (neither overwriting existing nor saving changes to a new workbook).

For more information, see Web Editing and Web Authoring.

Configure Projects, Groups, and Permissions for Managed Self-Service

Tableau Online and Tableau Server each provide an environment for easy open publishing and collaborative analysis of visualizations created in Tableau Desktop or web authoring. With that flexibility comes the challenge of making sure the right content is easy to find for the people who rely on it for their work. Likewise, making sure the access you allow doesn’t create performance or management nightmares on the site.

To address these challenges, many administrators set up their Tableau sites for what we’ll refer to as managed self-service. This is just a way of saying that the site allows areas of open collaboration and web editing, alongside areas in which access to data and reports is more controlled. As the site administrator, you put guidelines in place to help users figure out where to go for the type of work they need to do.

To get started with a managed self-service approach, the following sections discuss how you as the site administrator can meet the following objectives:
Create projects on the Tableau Server or Tableau Online site to match the ways people need to work with content.

For example, some projects are open to all for collaboration; others are visible only to authorized publishers.

Create user groups based on the type of access users need to the content.

Create a clear and scalable permissions strategy.

Note: The information provided here is adapted and simplified from practices of existing Tableau Zen Masters and customers who have shared their experiences. Links to their talks are available at the bottom of this page.

Create a project team and adopt a permissions strategy

Although changing the project structure on your site after your users are publishing to it is not impossible, it’s difficult and can be daunting. So before you make any lasting decisions or take definitive actions on your Tableau site, we recommend that you recruit users from various segments of your Tableau population, to create a project team of people who have differing uses for Tableau content.

Your permissions strategy will help your environment scale as you add new Tableau users. Make sure it incorporates two important practices: manage permissions only for groups, and set permissions only at the project level. Setting permissions at the individual user level and on individual content resources becomes unmanageable quickly. If you need to deviate from this practice, make sure you document and communicate your strategy to other administrators and project leaders.

Steps to coordinate projects and groups

To get projects and permissions (content) to work together with groups (people) in a managed self-service environment, you generally take the following steps:
Tableau Server on Linux Administrator Guide

1. Plan your permissions: Find common themes in the type of access users need. This helps determine projects and groups.

2. Remove permissions that will cause ambiguities

3. Create groups

4. Assign permissions to the groups

5. Create projects and adjust permissions

6. Lock permissions in each project

If you decide to follow the guidelines described here, you might want to Automate working with groups and projects.

1. Plan your permissions

Before you create groups and start assigning permissions, create a list of people who need access to content, and arrange them in groups according to what they’ll want to do.

For example, someone who publishes or moves a data source to a certified content project would need different level of access than someone who only consumes published reports. (We use the term “certified” to mean “trusted” — these are the data sources or reports that your Tableau community can trust to be a source of truth for your organization.)

Keep in mind also that you can set permissions differently for each project. So someone who is a data steward for the Ops department might not get the equivalent access to the Marketing content.

This exercise, done outside of the Tableau environment, can be the most challenging part of setting up a site.
Use a closed permissions model for managed content

General models for setting permissions are open or closed. In an open model, users get a high level of access, and you explicitly deny capabilities. This model can work when your organization is very small, and everyone has a similar level of responsibility.

In a closed model, users get only the access they need to do their jobs. This is the model security professionals advocate, and the examples in this article will attempt to show.

2. Remove permissions that will cause ambiguities

Every site has a Default project and an All Users group. Any user added to the site becomes a member of the All Users group automatically. The Default project works as a template for new projects in the site and cannot be deleted, but you can change the permissions. Creating groups and setting baseline permissions here helps you to know and manage exactly who gets what level of access for each new project.

In the managed self-service context, setting baseline permissions means removing the permissions from the All Users group, so that the permissions are enabled only on groups you create and have control over.

1. Select the Content tab to open the top-level projects on the site.

2. On the Default project’s Actions (…) menu, select Permissions.
3. Next to the **All Users** group name, select ..., and then select **Edit**.

4. In the drop-down lists under **Project**, **Workbooks**, and **Data Sources**, select **None**.

5. Select **Delete** to apply the changes.
3. Create groups

You create groups to match what people need to do with a set of content. In this case “a set of content” refers to the workbooks and data sources in a project.

When you create your groups, use descriptive names that make sense for your organization. For example, one possible set of groups might be as follows:

- **Project leaders.** You might also think of these as project-level administrators. Users who can perform all available capabilities on data sources, with the possible exception of setting permissions on them. People in this group can be site administrators, or users whose job it is to approve or certify data models or reports.

To grant administrator capabilities at the project level, you can assign the *Project Leader* permissions role to users with the appropriate site roles. To learn more, see Project administration.

- **Analysts/Publishers.** This group is for users who can publish workbooks to production and other open projects, use web editing on some projects, and connect to data sources certified by the data stewards. This group is not allowed to set permissions on content or move it between projects.

- **Business Users.** This group is the most likely to include people who do not use Tableau Desktop, but use data to answer questions and make business decisions. They can view and interact with workbooks only in specific projects, and they can’t publish, edit, save, or delete anything.

- **Administrators.** Depending on the size of your deployment, managing site or server administrators as a group helps you keep track of who has that level of access.

**Note:** Users with the Server Administrator or Site Administrator Creator site role have access to everything on the site, regardless of the groups you add them to.
If you have multiple Tableau roles per department, creating corresponding groups manually can be labor intensive. For alternatives, see Automate working with groups and projects later in this article.

**Learn more:** Add Users to a Group

4. Assign permissions to the groups

After you create groups, you can assign permissions in one of the following ways:

- In the Default project, apply a core set of permissions on each group that will stay more or less the same for all projects. You can then make minor adjustments in specific projects.

  Or

- Keep the Default project clean, and apply permissions only on projects you create.

For the example we’re using, it makes more sense to set permissions templates in the Default project. You will want to explicitly deny some capabilities across the board, and then allow them on only a few projects where you want to allow more open access.

Set permissions templates

1. While you have the Default project open, on the **Actions** menu (…), select **Permissions**.

   The **Permissions** pane shows only the All Users group that has no permissions.

2. Create a permission rule for each group as follows:

   a. Select **Add a user or group rule**, and then select one of your groups.

      This adds the group to the **User/Group** column, open for editing.

   b. Select a permission role in the **Project, Workbook, and Data Source**
Permission roles are predefined sets of capabilities that make setup easier.

c. Refine permissions in any of the columns by selecting the expand icon (►) to display individual capabilities and set them explicitly.

3. **Lock permissions to the project.**

Example

For the groups defined in 3. Create groups, here is one way you might set default permissions.

**Project leader roles**

- Project: **Project Leader**
- Workbooks: **Editor**
- Data Sources: **Editor**

This gives site administrators and data stewards full access to a project and its content. If you’re an IT admin, this enables you to delegate Tableau content administration to people who are closer to that content.

**Analyst Publisher roles**

- Project: **Publisher**
- Workbooks: **Editor**
- Data Sources: **Connector**

**Business Users roles**

- Project: **Viewer**
- Workbooks: **Interactor**
- Data Sources: **Denied**

**Default project settings for individual capabilities**
Under Workbooks, set **Web Edit** and **Download Full Data** to **Deny**.

This assumes you want to allow web editing and downloading data only on select projects. When you create those projects, you can refine the permissions.

- If you want to put more than a couple of users in the Project administrators group for each project, consider denying the **Set Permissions** capability for that group. An alternative for delegating the task of setting permissions is to set individual users’ site roles to Site Administrator.

- Leave capabilities in the Edit category set to **None**.

- If you want to allow other capabilities only as an exception, set those to **Deny** here as well.

5. Create projects and adjust permissions

After the Default project is set with your custom permissions template, you can create projects that allow the content use cases you identified. For each project, you can adjust the default permissions as appropriate.

**Example project structure**

One way to structure projects could be to reflect the following use cases:

**Workbooks shared for open collaboration on the server**
Anyone in the department can publish to the open-collaboration project while their content is in development. Colleagues can collaborate using web editing on the server. Some people call this a sandbox, some call it staging, and so on. On this project you can allow web editing, saving, downloading, and so on.

Here you want not only to enable collaboration, but also to enable people who don’t have Tableau Desktop to contribute and provide feedback.

**Shared reports that cannot be edited**

This could be a project that people who create workbooks and data sources (Analysts and Data Stewards) could publish to when they want to make content available to business users for viewing, with confidence that their work cannot be “borrowed” or modified.

For this type of project, you would deny all capabilities that allow editing or getting the data off of the server for reuse. You would allow viewing and interacting capabilities.

**Vetted data sources for Analysts to connect to**

This would be where Data Stewards publish the data sources that are meet all of your data requirements and become the “source of truth” for your organization. Project leaders on this project can certify these data sources, so that they rank higher in search results and are included in recommended data sources.

You would allow authorized Analysts (that is, the Publishers group described earlier) to connect their workbooks to data sources in this project, but not download or edit them. You would deny capabilities to the Business Users group, so those users would not even see this project.

**Inactive content**

Another possibility is to segregate workbooks and data sources that the site’s administrative views show haven’t been used for a period of time. You could give content owners a time limit before their content is removed from the server.
Whether you do this or delete directly from the working projects is up to your organization. In an active environment, don't be afraid to be intentional about removing content that is not being used.

**Source for workbook templates**

This is a project that people can download from but not publish or save to, where authorized publishers or project leaders make template workbooks available. Templates that have your organization’s approved fonts, colors, images, and even data connections built in can save authors a lot of time and keep your reports looking consistent.

Help project leaders manage content and users find it

- Devise a scalable project-naming scheme that makes sense in your organization.
  
  For example, basic structure might be `<Department> - <ContentUse>`; such as **Ops - Production**.

- Use the project’s **Description** field.

  The description you enter when you create a project appears when you hover the pointer over the project thumbnail, as well as on the **Project details** page.
6. Lock permissions in each project

After you refine the capabilities for each group in a project, you can lock the project’s permissions. Do this on the Default project, too.

1. With a project’s Permissions page open, select the button next to Permissions for workbooks and data sources are

2. In the dialog box that appears, select Locked to the project.

Locking permissions prevents publishers from setting permissions explicitly as part of the publishing process in Tableau Desktop. Instead, content inherits permissions set on the project it’s published to, and only administrators and project leaders can set permissions.

Automate working with groups and projects

Creating multiple groups and projects and setting permissions manually can get a little tedious. To automate these processes, as well as make them repeatable for future updates, you can perform these tasks using REST API commands.

You can use tabcmd commands for tasks such as adding or deleting a single project or group and adding users, but not for setting permissions.
Next steps

Besides projects, groups, and permissions, other data governance themes include:

**User education**

Help *all* of your Tableau users become good data stewards. The most successful Tableau organizations create Tableau user groups, have regular training sessions, and so on.

For a common approach to orienting users to the site, see Dashboard-based Custom Portals.

For publishing and data certification tips, see the following topics:

- Use Certification to Help Users Find Trusted Data
- Prepare for Publishing a Workbook (links to Tableau Help)
- Best Practices for Published Data Sources (links to Tableau Help)

**Optimize extract refresh and subscription activity**

If you use Tableau Server, create policies for extract refresh and subscription schedules, to avoid them dominating the site’s resources. The TC customer presentations by Wells Fargo and Sprint address this subject in detail. In addition, see the topics under Performance Tuning.

If you use Tableau Online, see the following topics to become familiar with the ways people can refresh extracts:

- Keep Data Fresh
- Use Tableau Bridge to Expand Data Freshness Options

**Monitoring**

Use administrative views to keep an eye on the site’s performance and content use.

Administrative Views
Learn how Tableau and some of our customers address governance and self-service

The following list contains links to data governance and Center of Excellence (COE) presentations given at the Tableau Conference over recent years. Even if Tableau versions have evolved, the principles remain the same. You can explore the playlists for other videos related to COE, managing Tableau at scale.

Creating a Centre of Excellence in Tableau (TC Europe 2018)

Server Admins: Don't Fear Web Authoring (Sprint, TC16)

The Past, Present, & Future at Charles Schwab (TC 17)

Content Strategies in Tableau (TC 17)

Use Projects to Manage Content Access

When Tableau Desktop users publish a workbook or data source to a site on Tableau Server, they can select a project to publish it to. If they don’t specify a project, their content is published to the Default project.

As an administrator, you can create projects to hold and organize related content resources, or to delegate content management. Content resources on Tableau Server are workbooks, views, and data sources, and the projects that hold them.

The following image shows content within the top-level Operations project in the web authoring environment. The Operations project contains a few child projects (highlighted) and published workbooks. A project can also contain other content types, such as data sources and flows.
Why use projects

Projects help you to create a scalable process for managing access to the content published to Tableau Server. Advantages they have include:

- They enable administrators to delegate content management to project leaders who work with the content more closely, without having to give them administrator access to site or server settings.

  Project leaders can create nested projects under their top-level project, enabling them to maintain their team’s content within a single hierarchy.

  **Note:** Project owners can delete top-level projects they own.

- They can make the site easier to navigate for self-service users.
They segment the Tableau Server site into areas that give users access based on how they use the data published to those areas, or on the Tableau user group they work with.

You can hide projects from groups who don’t need to use them, create a distinguishable project-naming scheme, and take advantage of project descriptions to clarify how to use the project.

- They enable you to track permissions effectively.

You can create groups based on the level of content access users in the group need, and set default permissions on projects. This enables you to know exactly which capabilities new users get by default, and likewise which capabilities all users get when a new project is created.

When to create project hierarchies (example)

Many organizations have several or more distinct groups of Tableau users, each with its own priorities and leaders. These groups might share some organization-wide content (or even draw from an org-wide pool of data sources), but primarily they use data and reports that are specific to their team. In this or similar scenario, an example for using project hierarchies might look as follows:

1. You, as a site or server administrator, can create top-level projects for each of your distinct Tableau teams.

2. On each top-level project, you assign the Project Leader role to team leads, and change project ownership.

   Project leaders effectively are the content administrators, so it’s important that they understand how permissions work in Tableau, along with Tableau content management best practices.

3. Each project leader takes over their project, creating the structure within the project that works for their team.
That is, they can create child projects they need, based on how their team members collaborate and share data and reports.

The benefit to you as the site administrator is that you can focus on system health. The benefit to your Tableau users is that people who know the best practices for working with Tableau and data can manage these things for their teams, without having to submit IT requests to change permissions or add projects.

Why not use sites?

If you manage your own Tableau Server deployment, you can create as many sites as you want. However, for managing data and reports across your company, projects allow the flexibility you need to administer shared data and reports, and users who might belong to multiple groups. Many Tableau administrators configure projects as described in the previous section, to expose only what’s necessary to users who need to work with it. Projects work better than sites for evolving content from development to staging to production.

Sites work well when content can remain completely separate during all phases, and there is little to no user overlap. A good (and common) example for using multiple sites is to create a site for each of multiple external clients, whose published content you manage as a consultant or vendor. Our own Tableau Online is an example of this on a large scale. Another example might be to use a separate site for sensitive content that you want only specific Human Resources or medical staff to use.

Project-level administration

As a server or site administrator, you can delegate administration of projects and their content, without exposing access to your site or server settings. You can do this by changing the ownership of a project or granting a group or user the Project Leader permissions role on a project.

The project owner is always one individual user. By default, the user who creates a project is its owner. The project owner has administrative access to the project and content in it—including making someone else the owner and assigning Project Leader permissions.
The **Project Leader** permissions role provides a way to allow multiple users administrative access to a project, its child projects, and all workbooks and data sources in those projects.

A project leader does not have to be a project owner or administrator. In addition to server and site administrators, the full scope of Project Leader permissions is available to users with a **Creator** or **Explorer (Can Publish)** site role.

Project ownership and project leader access in project hierarchies

In a multi-level project hierarchy, a user or group that is given the Project Leader permissions role to a project, at any level within the hierarchy, is implicitly given Project Leader access to all of that project’s child projects and their content items.

To remove the Project Leader access, you must do so at the parent level in the hierarchy on which the ownership or role was explicitly assigned.

Similarly, the owner of a project at any level has project leader access to all content in that project, as well as to any of its child projects, even if they do not own the child projects.

Only a project owner or administrator can change ownership of a content resource, and this can be done regardless of whether the project permissions are locked at the top level.

Actions project-level administrators can take on projects

Project leaders and owners can perform the tasks in the following list, as can server or site administrators.

- Create and delete projects as follows:

  Server or site administrators can create or delete top-level or nested projects anywhere on the site.

  Project owners and project leaders can create and delete child (nested) projects in projects they own, or on which they have Project Leader permissions.

- Project owners can change ownership of their projects. Both project owners and
project leaders can assign the Project Leader permissions to groups or users.

As a project leader or owner, if you assign the Project Leader permissions role to someone else, remember that full access to this role’s capabilities depends on the user’s site role, as specified earlier in this section.

- Set permissions for a project, as well as the child projects, workbooks, and data sources in it.

- Lock permissions to apply the top-level project’s default settings to all workbooks, data sources, and child projects and their content. Permissions can be modified only from the project on which they’re locked.

In a locked project hierarchy, only the owner or project leader of the top-level project in the hierarchy can change permissions.

- Move workbooks and data sources to another project to which they have Project Leader or owner access.

  Moving the project can affect permissions. For information, see Move content.

- Run, add, or remove extract refresh schedules.

See also Project permissions.

How the Default project acts as a permissions template

Tableau creates a Default project with every site.

The Default project serves as a template for new top-level projects you create on the site. When you create a new top-level project, settings and permissions from the Default template are applied to the new project, including permissions set on content within the project.

**Note:** Nested projects (projects you create within other projects) take the permissions set at their parent project, not the Default project.
Before you create other top-level projects, you can take steps to set up the Default project, to help you to know exactly:

- Which type of user gets what level of access for each new project.
- How you might need to modify permissions for each new project.

For more information, see Steps to coordinate projects and groups in the topic Configure Projects, Groups, and Permissions for Managed Self-Service.

Prevent publishers from changing permissions on content they own

Administrators and project leaders can prevent users from changing the permissions for workbooks and data sources in a project hierarchy. For example, you can disable the option to set permissions during the publishing process, and prevent publishers and content owners from changing them after publishing. To do this, you lock content permissions to the project. For more information, see the links below.

Add Projects and Move Content Into Them

A content resource (workbooks and data sources) can live in only project. Server and site administrators can add or remove top-level projects on a site, and move published content from one project to another. Project leaders with appropriate site roles can add or remove child projects and move content between projects on which they have Project Leader access.

This article contains the steps for creating and moving projects. We recommend becoming familiar with the following related content as well:

- To learn about projects and when or why to use them, see Use Projects to Manage Content Access.
- Before you create project hierarchies, become familiar with Project permissions.
- To see the specific site roles that allow full Project Leader access, see Project-level administration.
Add a top-level or child (nested) project

1. While you’re signed in to Tableau Server as an administrator or project leader, select the Content tab, and then do one of the following:
   - Select Create > Project to create a new top-level project (only administrators can do this).
   - Navigate to and open the project in which you want to create a sub-project, and then select Create > Project.

   If you’re not sure where to find a child project, display filters, and select Show all projects.

2. Enter a name and description for the project, and then click Create.

   ![New Project](image)

   You can include formatting and hyperlinks in the project description. Select Show formatting hints for syntax.

   **Note:** To edit a project description later, select it to open it, select the information icon next to its name, and then click Edit.
Move a content resource to another project

1. On the **Content** tab, find the content resource you want to move.

   If you're not sure where to find a child project, display filters, and select Show all projects.

   For other content types, you can navigate through its project hierarchy, or by selecting the content type on the **Explore** menu.

2. On the workbook's **Actions(...)** menu, select **Move**.

3. Select the new project for the workbook, and then click **Move Content**.

   Moving a project includes moving everything in it, including child projects and their content.
How moving projects affect Project Leader permissions

When you move a project, Project Leader permissions adapt to the new project environment.

- When the target project hierarchy is **locked**, previous Project Leader permissions are removed, and new Project Leader permissions are granted according to those set at the top-level of the target hierarchy.

- When the target project hierarchy is **unlocked** (managed by owner), previous implicitly granted Project Leader permissions are removed, explicitly set Project Leader permissions are retained, and new Project Leader permissions are granted according to those set at the top-level of the target hierarchy.

Delete a project

When you delete a project, all of the workbooks and data sources in the project are also deleted from the site. If you want to delete a project but not its content, move the content to another project, and then delete the project.

**Important**

- You cannot undo deleting a project.

- Deleting a project deletes all content in it, including child projects and their content.

- You cannot delete the **Default** project.

To delete a project:

1. On the **Content** tab, find the project you want to remove.

   If you’re not sure where to find a child project, display filters, and select **Show all projects**.

2. On the project’s **Actions (…)** menu, select **Delete**.

3. Confirm that you want to delete the project.
Required access level for moving content

Moving content is effectively like removing it from one project and publishing it to another. For non-administrators, the permissions needed on the source project are different than those needed on the destination project.

**Note:** The term *permissions role* refers to the collection of related capabilities you can assign as a set.

![Permissions Role](image)

Required site role

To move content, users must have one of the following site roles:

- Server Administrator (Tableau Server only)
- Site Administrator Creator or Site Administrator Explorer
- Creator or Explorer (Can Publish)

Users with a Server Administrator or Site Administrator site role do not need any additional capabilities.

Required permissions for the project that users move content **to**

Non-administrators must have the Publisher or higher permissions role set on the project that is the move destination.
Required permissions for the project that users move content from

Non-administrator users must have these additional capabilities on the project:

- Ownership of the project or the Project Leader or Publisher permissions role.

**OR**

- In a project hierarchy, project owner of, or Project Leader permissions role on the project or a parent project higher in the hierarchy.

If they have the Publisher permissions role on the project, and they are not the project owner (or owner or leader of a parent project), they must have the following capabilities:

**Data sources:** Data source owner

**Workbooks:** Workbook owner or Move capability set explicitly on the workbook.

You can grant the Move capability individually, or along with related capabilities allowed through the Editor permissions role.

**Add a Project Image**

To help distinguish projects you manage on Tableau Server (and help your users find them), you can add an image that appears in the thumbnail. Your image must meet the following requirements:

- The image must be accessible using the HTTP or HTTPS protocols. Shared network directory and related protocols (UNC, SMB, AFP, NFS, etc) are not supported.
- All users who access the project must have, at a minimum, "read-only" permission on the target image.
- The image must be common internet format: .jpg, png, or gif.
Set a project image

1. Sign in to a site on Tableau Server. In the list of **Top-level Projects** you have access to, select or navigate to the project you want to update. In this example, we’ll add an image to the Statistics project folder.

   If you’re not sure where to find a child project, use the **Explore** drop-down list and select **All Projects**.

2. Click the **Details** icon (i), to open the **Project details** dialog box, and then click **Edit**.
3. In the **About** field, you can enter a description for your project (optional), for example "Global and US statistics." At the end of the project description, add the URL for your image using the following syntax:

```
!http://www.example.com/image.png!
```
Select **Show formatting hints** to see how you can format description text.

**Note:** Images embedded in project descriptions cannot be resized or positioned. Recommended size is (300 x 184 pixels). Images that are not 300 x 184 pixels may be stretched, shrunk, or cropped to fit the width of the thumbnail. In addition, they must be added at the end of the project description and be enclosed in ! (exclamation marks), otherwise they will not be displayed as the thumbnail.

4. Click **Save**.
Permissions

Permissions define what capabilities a user is allowed or denied, controlling what they can see and do with content such as workbooks and data sources.

Permissions are set in the Permissions dialog box. At the top, permission rules configure capabilities for groups or users as allowed, denied, or unspecified. Below, the permissions grid displays the effective permissions for users.
Permissions fundamentals

Tableau sites use projects to organize content and groups to organize users. Permissions are applied to content or projects and determine how users or groups interact with pieces of content.

Permissions are made up of capabilities, the ability to do things like view content, web edit, download data sources, or delete content. Permission rules establish what capabilities are allowed or denied for a user or group on a piece of content. The exceptions are administrators, who have all capabilities on all content, and content owners, who have all capabilities on their own content.
For each user, there must be a final resolution of allowed or denied for each capability, known as *effective permissions*. The interplay between license level, site role, and permission rules also factor into the final determination of what a user can or cannot do. To ensure users have the correct access to content, it’s important to understand all the pieces that go into the final effective or resulting permissions for a user. For more information see Effective permissions.

Some tasks such as creating new workbooks from a browser (web authoring) or moving content might require specific configurations of several capabilities rather than being captured in a single capability.

**Note:** When talking about permissions in general, it’s common to see a phrase like “a user must have the delete *permission*”. This is easy to understand in a broad context. However, when working with permissions at a technical level like in this article, it’s more accurate to say “the delete *capability*”. In this topic we’ll use the more precise term *capability*, but you should be aware that you might see *permission* in other places.

Permissions, site roles, and licenses

Adding a user to a Tableau Server requires a license. For each site the user belongs to they have exactly one site role, restricted by their license. A user has permissions for content on the site, restricted by what their site role allows. Licenses and site roles apply to users. Permission capabilities apply to content.
License levels determine the maximum site role a user can have on that server.

- Server Administrator, Site Administrator Creator, and Creator site roles require a Creator license.
- Site Administrator Explorer, Explorer (can publish), and Explorer site roles require at least an Explorer license.
- Viewer site role requires at least a Viewer license.

For Tableau Server, a user consumes only one license per server, even if they are a member of multiple sites. If a user is a member of multiple sites, their required license level is determined by their highest site role. (For example, if a user has a Creator site role in one site and a Viewer site role in two others, they must have a Creator license.)

Site roles are assigned to a user when the user is created, and again any time they are added to another site. Users have a site role for each site they are a member of.

- Site roles determine the maximum capabilities a user can have in that site. (For example, a user with a site role of Viewer will never be able to download a data source even if that capability is explicitly granted to them on a specific data source.)
- Site roles do not inherently grant any capabilities in and of themselves—with the exception of the administrator site roles. Administrators always have all capabilities applicable to their license level.

Permissions consist of capabilities, things like the ability to save to a project, web edit a workbook, connect to a data source, etc. They apply to group or user on a specific piece of content (project, data source, workbook, view, or flow).

- Permission capabilities are not given to a group or user in a vacuum but rather in the context of content. A user can have different capabilities for different content assets.
- Permissions are evaluated based on the interplay of a user’s site role and the permission rules for that user or any groups they are members of.
- Some actions such as web authoring might require combinations of capabilities.

Site Roles and their maximum capabilities

These tables indicate what capabilities are available to each site role. There may be other
ways for a user with a site role to perform an action. For example, although Viewers cannot be given the **Share Customized** capability, they can share views by copying the URL. See General capabilities allowed with each site role for more information on what each site role can do.

### Projects

<table>
<thead>
<tr>
<th>Capability</th>
<th>Creator</th>
<th>Explorer (can publish)</th>
<th>Explorer</th>
<th>Viewer</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>View</strong></td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td><strong>Save</strong></td>
<td>✔️</td>
<td>✔️</td>
<td>✗</td>
<td>✗</td>
</tr>
<tr>
<td><strong>Project</strong></td>
<td>✗</td>
<td>✔️</td>
<td>✗</td>
<td>✗</td>
</tr>
</tbody>
</table>

### Workbooks

<table>
<thead>
<tr>
<th>Capability</th>
<th>Creator</th>
<th>Explorer (can publish)</th>
<th>Explorer</th>
<th>Viewer</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Save</strong></td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td><strong>Download</strong></td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td><strong>Image/PDF</strong></td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td><strong>Summary Data</strong></td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td><strong>View Comments</strong></td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
</tbody>
</table>
### Tableau Server on Linux Administrator Guide

<table>
<thead>
<tr>
<th>Action</th>
<th>Creator</th>
<th>Explorer (can publish)</th>
<th>Explorer</th>
<th>Viewer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add Comments</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Filter</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Download</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Full Data</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Share Customized</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Web Edit</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Save</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Download</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Workbook/Save As</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

*Although the Explorer role can be given the **Move** capability, they cannot have the **Save** capability on a project and therefore there is no place for them to move content to. The **Move** capability should therefore be considered not possible for Explorer site roles.*
Tableau Server on Linux Administrator Guide

<table>
<thead>
<tr>
<th>Capability</th>
<th>Creator</th>
<th>Explorer (can publish)</th>
<th>Explorer</th>
<th>Viewer</th>
</tr>
</thead>
<tbody>
<tr>
<td>View</td>
<td>![Checkmark]</td>
<td>![Checkmark]</td>
<td>![Checkmark]</td>
<td>![Checkmark]</td>
</tr>
<tr>
<td>Connect</td>
<td>![Checkmark]</td>
<td>![Checkmark]</td>
<td>![Checkmark]</td>
<td>![Checkmark]</td>
</tr>
<tr>
<td>Save</td>
<td>![Checkmark]</td>
<td>![Checkmark]</td>
<td>![X]</td>
<td>![X]</td>
</tr>
<tr>
<td>Download</td>
<td>![Checkmark]</td>
<td>![Checkmark]</td>
<td>![Checkmark]</td>
<td>![X]</td>
</tr>
<tr>
<td>Data Source</td>
<td>![Checkmark]</td>
<td>![Checkmark]</td>
<td>![X]</td>
<td>![X]</td>
</tr>
<tr>
<td>Delete</td>
<td>![Checkmark]</td>
<td>![Checkmark]</td>
<td>![X]</td>
<td>![X]</td>
</tr>
<tr>
<td>Set Permissions</td>
<td>![Checkmark]</td>
<td>![Checkmark]</td>
<td>![X]</td>
<td>![X]</td>
</tr>
</tbody>
</table>

Flows

Note that Flows are part of the Data Management Add-on.

<table>
<thead>
<tr>
<th>Capability</th>
<th>Creator</th>
<th>Explorer (can publish)</th>
<th>Explorer</th>
<th>Viewer</th>
</tr>
</thead>
<tbody>
<tr>
<td>View</td>
<td>![Checkmark]</td>
<td>![Checkmark]</td>
<td>![Checkmark]</td>
<td>![Checkmark]</td>
</tr>
<tr>
<td>Run Flow</td>
<td>![Checkmark]</td>
<td>![Checkmark]</td>
<td>![X]</td>
<td>![X]</td>
</tr>
<tr>
<td>Save</td>
<td>![Checkmark]</td>
<td>![Checkmark]</td>
<td>![X]</td>
<td>![X]</td>
</tr>
<tr>
<td>Download</td>
<td>![Checkmark]</td>
<td>![Checkmark]</td>
<td>![Checkmark]</td>
<td>![X]</td>
</tr>
<tr>
<td>Flow</td>
<td>![Checkmark]</td>
<td>![Checkmark]</td>
<td>![X]</td>
<td>![X]</td>
</tr>
<tr>
<td>Move</td>
<td>![Checkmark]</td>
<td>![Checkmark]</td>
<td>![X]</td>
<td>![X]</td>
</tr>
<tr>
<td>Delete</td>
<td>![Checkmark]</td>
<td>![Checkmark]</td>
<td>![X]</td>
<td>![X]</td>
</tr>
<tr>
<td>Set Per-</td>
<td>![Checkmark]</td>
<td>![Checkmark]</td>
<td>![X]</td>
<td>![X]</td>
</tr>
</tbody>
</table>
*Although the Explorer role can be given the Move capability, they cannot have the Save capability on a project and therefore there is no place for them to move content to. The Move capability should therefore be considered not possible for Explorer site roles.

Data Roles

Note that Data Roles are part of the Data Management Add-on.

<table>
<thead>
<tr>
<th>Capability</th>
<th>Creator</th>
<th>Explorer (can publish)</th>
<th>Explorer</th>
<th>Viewer</th>
</tr>
</thead>
<tbody>
<tr>
<td>View</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Save</td>
<td>✓</td>
<td>✓</td>
<td>❌</td>
<td>❌</td>
</tr>
<tr>
<td>Move</td>
<td>✓</td>
<td>✓</td>
<td>*</td>
<td>❌</td>
</tr>
<tr>
<td>Delete</td>
<td>✓</td>
<td>✓</td>
<td>❌</td>
<td>❌</td>
</tr>
<tr>
<td>Set Per-missions</td>
<td>✓</td>
<td>✓</td>
<td>❌</td>
<td>❌</td>
</tr>
</tbody>
</table>

*Although the Explorer role can be given the Move capability, they cannot have the Save capability on a project and therefore there is no place for them to move content to. The Move capability should therefore be considered not possible for Explorer site roles.

Set permissions

Permissions can only be established for existing users, groups, or content. Managing permissions is easier when permission rules are established for groups instead of individuals.
Set Project Permissions

For administrators and project leaders

Permissions can be set at the project level for both the project itself and for any content in the project. For example, if workbook permissions are configured at the project level, all workbooks published into that project inherit those default permissions. However, the Creator can choose to change the permissions during publishing, or certain users can change the permissions on published content. To enforce the permissions established at the project level, Content Permissions can be locked to the project. For more information, see Lock project permissions.

To set permissions at the project level:

1. Navigate to the project
2. Open the Actions menu (…) and click Permissions. The permissions dialog box opens.

This dialog box has two main areas: permission rules at the top and the effective permissions grid below. Each section (Project, Workbooks, Data Sources, Flows, Data Roles) can be expanded (►) to reveal the capabilities for that type of content.
With a row selected at the top, the effective permissions grid populates. Use this to verify permissions. Hovering over a capability square provides information about why the capability is allowed or denied for that specific user.

3. To modify an existing permission rule, open the Actions menu (…) for that row and click **Edit**.
4. To create a new rule,
   a. Select **Add a user or group rule**.
   b. If necessary, use the drop-down box to the right to change between groups and users.
   c. Select a group or user from the drop-down box. This creates a row where you can configure the permission rule.
5. In the row for the permission rule, choose an existing permission role template from the drop-down box for each section, or create a custom rule by expanding a section (►) and clicking the capabilities.
   One click sets the capability to **Allowed**, two clicks sets it to **Denied**, and a third click clears the selection (**Unspecified**).
6. When finished, click **Save**.

**Set Content Permissions**

*For administrators, project leaders, and content owners*

If project permissions are not locked, permissions for individual pieces of content can be modified.

**Warning:** Tableau recommends managing permissions at the project level within the Tableau site. These steps are relevant only for content in projects where permissions are managed by the owner.

Set permissions on content

1. Navigate to the content (workbook, data source, flow, data role)
2. Open the Actions menu (…) and click **Permissions**. The permissions dialog box opens.
This dialog box has two main areas: permission rules at the top and the effective permissions grid below.

With a row selected at the top, the effective permissions grid populates. Use this to verify permissions. Hovering over a capability square provides information about why the capability is allowed or denied for that specific user.

3. To modify an existing permission rule, open the Actions menu (...) for that row and click Edit.
4. To create a new rule,
   a. Select + Add a user or group rule.
   b. If necessary, use the drop-down box on the right to change between groups and users.
   c. Select a group or user from the drop-down box. This creates a row where you can configure the permission rule.
5. In the row for the permission rule, choose an existing permissions role template from the drop-down box or create a custom rule by clicking the capabilities.

One click sets the capability to Allowed, two clicks sets it to Denied, and a third click clears the selection (Unspecified).
6. When finished, click **Save**.

Set permissions on a view

In some situations, it may be valuable to specify permissions on a view independently from the workbook that contains it. To set permissions on a published view, navigate to the view within a published workbook and follow steps above.

**Warning**: While it is possible to set view-level permissions within a workbook, we strongly recommend managing permissions at the project (or workbook) level as much as possible. For views to inherit permissions, the project must be locked or the workbook must be published with **Show Sheets as Tabs**. See Show or Hide Sheet Tabs for more information.

Set Permissions on Publish

*For content publishers*

If project permissions are not locked, permissions for data sources or workbooks can be set when publishing from Tableau Desktop.

**Warning**: Tableau recommends managing permissions at the project level within the Tableau site. These steps are relevant only for content in projects where permissions are managed by the owner.

1. From the publish dialog box, click the Edit link for **Permissions**.
   
   If the Edit link is unavailable, permissions are locked to the project and can’t be modified except by the project owner, project leader, or an administrator.

2. The Add/Edit Permissions dialog box shows any existing permission rules. Click **Add** to add a new permission rule or **Edit** to modify an existing permission rule
   
   a. Select the group or user from the left pane. You can expand a group to see which users it contains.

   b. Use the selector at the top of the right pane to choose an existing template, or use the radio buttons to create a custom rule.
Note that effective permissions cannot be inspected from the publishing dialog box.

3. When finished, click OK and resume publishing.

**Note:** Permissions cannot be set while publishing flows from Tableau Prep Builder. To set permissions on a flow, refer to the steps for Set Project Permissions or Set Content Permissions.

**Tip:** By default, all users are added to an “All Users” group that has basic permissions for content. To start with a clean slate when building your own permission rules, we recommend that you edit the All Users group to remove any permissions (set the permission role template to None). This will help prevent any ambiguity down the road by reducing the number of rules that apply to any given user and therefore making effective permissions easier to understand.
Permission capabilities

Permissions are made up of capabilities, or the ability to perform a given action on a piece of content (project, workbook, view, data source, or flow). Each type of content has a series of permission role templates to make it easier to assign capabilities quickly, but these can be modified to create custom capability combinations.

Permission Roles

When working in a project, there are content types: Projects, Workbooks, Data sources, and—if you have the Data Management Add-on—Flows and Data Roles. Expand each section to see what capabilities are available.

All content has permission role templates for None (which sets all capabilities to unspecified) and Denied (which sets all capabilities to denied), as well as permission role templates that allow specific groups of capabilities (leaving the rest unspecified). Capabilities can also be granted or denied independently in a custom configuration.

To modify an existing permission rule:

1. Click the Actions menu (…) next to the group or user name then click Edit
2. If necessary, expand (++) each section to reveal its capabilities
3. Click a capability in the rule once to set it to Allowed, twice to set it to Denied, or three clicks to clear it back to Unspecified
4. Click Save when you are done (or Delete, if setting all capabilities to Unspecified)

Projects

Projects have three capabilities, View, Save, and Project Leader.

View  Save  Project Leader

The Viewer permission role grants the View capability.

The Publisher permission role grants the View and Save capabilities.

The Project Leader permission role grants the Project Leader capability.
Details on project capabilities and permission roles

The image above shows a row for each of the permission role templates, illustrating which capabilities they grant or deny.

Capabilities:

- **View** allows a user to see the project. If a user has not been granted the view capability, the project does not appear for them. Note that granting the view capability for a project does not imply a user can see content in the project. This capability is only for the project itself.

- **Save** allows a user to publish content to the project from Tableau Desktop or Tableau Prep Builder. The save capability is also required to save content to the project from web authoring. (Saving is the web editing equivalent of publishing).
**Project Leader** allows a user to manage the project, including: setting permissions on all content or the project itself, locking project permissions, changing content ownership, moving content, and running refresh schedules.

In a project hierarchy, a group or user with the project leader capability on a project will have that capability for any child projects nested in that project. Removing the project leader capability can only be done at the highest level where it applies.

Permission roles:

**Viewer** allows the user or group to connect to the data source on the server.

**Publisher** allows the user or group to connect to, download, delete, and set permissions on data sources on the server. They can also publish data sources, and as long as they are the owner of a data source they publish, they can update connection information and extract refresh schedules. (The latter two capabilities are no longer available if an administrator or project leader changes data source ownership.)

**Workbooks**

Workbooks have three groupings of capabilities: View, Interact/Edit, and Edit.

- **View**
  - View
  - Download Image/PDF
  - Download Summary Data
  - View Comments
- **Interact/Edit**
  - Filter
  - Download Full Data
  - Share Customized
  - Web Edit
- **Edit**
  - Save
  - Download Workbook/Save As
  - Move
  - Delete
  - Set Permissions
The **Viewer** permission role grants all View capabilities.

The **Interactor** permission role grants all View and Interact/Edit capabilities.

The **Editor** permission role grants all View, Interact/Edit, and Edit capabilities.

**Note:** In a workbook that is configured to hide tabs, views (sheets, dashboards, stories) inherit the workbook permissions at publication, but any changes to permission rules must be made on individual views. We recommend showing sheet tabs whenever possible so views inherit their permissions from the workbook. See Show or Hide Sheet Tabs.

View capabilities are the same as those for workbooks, except for **Save**, **Download Workbook/Save As**, and **Move** which are only available at the workbook level.

**Details on workbook capabilities and permission roles**

```
<table>
<thead>
<tr>
<th>Permissions</th>
<th>View</th>
<th>Interact/Edit</th>
<th>Edit</th>
</tr>
</thead>
<tbody>
<tr>
<td>Managed by the owner</td>
<td>![Managed by the owner icon]</td>
<td>![Managed by the owner icon]</td>
<td>![Managed by the owner icon]</td>
</tr>
<tr>
<td>None</td>
<td>![None icon]</td>
<td>![None icon]</td>
<td>![None icon]</td>
</tr>
<tr>
<td>Editor</td>
<td>![Editor icon]</td>
<td>![Editor icon]</td>
<td>![Editor icon]</td>
</tr>
<tr>
<td>Interactor</td>
<td>![Interactor icon]</td>
<td>![Interactor icon]</td>
<td>![Interactor icon]</td>
</tr>
<tr>
<td>Viewer</td>
<td>![Viewer icon]</td>
<td>![Viewer icon]</td>
<td>![Viewer icon]</td>
</tr>
<tr>
<td>• Viewer</td>
<td>![Viewer icon]</td>
<td>![Viewer icon]</td>
<td>![Viewer icon]</td>
</tr>
<tr>
<td>• Interactor</td>
<td>![Interactor icon]</td>
<td>![Interactor icon]</td>
<td>![Interactor icon]</td>
</tr>
<tr>
<td>• Editor</td>
<td>![Editor icon]</td>
<td>![Editor icon]</td>
<td>![Editor icon]</td>
</tr>
<tr>
<td>• None</td>
<td>![None icon]</td>
<td>![None icon]</td>
<td>![None icon]</td>
</tr>
<tr>
<td>• Denied</td>
<td>![Denied icon]</td>
<td>![Denied icon]</td>
<td>![Denied icon]</td>
</tr>
</tbody>
</table>
```
The image above shows a row for each of the permission role templates, illustrating which capabilities they grant or deny.

Capabilities:

- **View** allows a user to see the workbook or view. If a user has not been granted the view capability, the workbook will not appear for them.

- **Download Image/PDF** allows a user to download each view as a PNG, PDF, or PowerPoint.

- **Download Summary Data** allows a user to view the aggregated data in a view, or in the marks they’ve selected, and download that data (as a .CSV).

- **View Comments** allows a user to view the comments associated with the views in a workbook.

- **Add Comments** allows a user to add comments to views in a workbook.

- **Filter** allows a user to interact with filters in the view, including keep only and exclude filters. Users lacking this capability will not see filter controls in the view.

- **Download Full Data** allows a user to view the underlying data in a view, or in the marks they’ve selected, and download that data (as a .CSV).

- **Share Customized** allows a user to shared saved customizations made to the view (such as filters and selections). These customizations appear as options for other users, they do not change the default state of the view for others.

- **Web Edit** allows a user to edit the view in a browser-based authoring environment. Note that creating new content in the browser or saving views from the web edit interface (Web Editing and Web Authoring) requires a specific combination of capabilities.
The Web Editing feature must also be enabled for the entire site or even users with this capability allowed will not be able to web edit. For more information, see Set a Site’s Web Authoring Access and Functions.

Save allows a user to overwrite the content asset on the server. When allowed, the user can re-publish a workbook, data source, or flow, or save (overwrite) a workbook in web authoring, thereby becoming the owner and gaining access to all permissions. Subsequently, the original owner’s access to the workbook is determined by their permissions just like any other user.

Download Workbook/Save As allows a user to download a packaged workbook (as a .twbx). Allows a user to save (publish) a copy from the web edit interface as a new workbook.

Move allows a user to move workbooks between projects. For more information, see Move content.

Delete allows a user to delete the workbook.

Set Permissions allows a user to create permission rules for the workbook.

Permission roles:

Viewer allows the user or group to view the workbook or view on the server.

Interactor allows the user or group to view the workbook or view on the server, edit workbook views, apply filters, view underlying data, export images, and export data. All other permissions are inherited from the user’s or group’s project permissions.

Editor sets all capabilities for the rule to Allowed.

Data Sources

Data sources have two groupings of capabilities, Use and Edit.
The **Connector** permission role grants all Use capabilities.

The **Editor** permission role grants all Use and Edit capabilities.

**Details on data source capabilities and permission roles**

The image above shows a row for each of the permission role templates, illustrating which capabilities they grant or deny.

**Capabilities:**

- **View** allows a user to see the data source on the server
**Connect** allows a user to connect to a data source in Tableau Desktop, Tableau Prep Builder, or web editing.

**Note**: If a workbook author embeds their credentials to a published data source in a published workbook, they are essentially embedding their **Connect** capability. Therefore, users can see the data in the workbook regardless of their own **Connect** capability for that data source. If the workbook author does not embed their credentials to the published data source, the user needs their own **Connect** capability to the data source in order to consume the workbook. For more information, see Data access for published Tableau data sources.

**Save** allows a user to publish data sources to the server and overwrite data sources on the server.

**Download Data Source** allows a user to download the data source from the server (as a .tdsx)

Cube data sources, like those for Microsoft Analysis Services or Oracle Essbase connections, must be used locally. To download the published data source to Tableau Desktop, the user must have the Download capability. For more information, see Cube Data Sources.

**Delete** allows a user to delete the data source.

**Set Permissions** allows a user to create and edit permission rules for the data source.

Permission roles:

**Connector** Allows the user or group to connect to the data source on the server.

**Editor** Allows the user or group to connect to, download, delete, and set permissions on data sources on the server. They can also publish data sources, and as long as they are the owner of a data source they publish, they can update connection information and extract
refresh schedules. (The latter two capabilities are no longer available if an administrator or project leader changes data source ownership.)

Flows and Data Roles

*Note that Flows and Data Roles are part of the Data Management Add-on.*

Flows have two groupings of capabilities: Run and Edit.

Run
- View
- Run Flow

Edit
- Save
- Download Flow
- Move
- Delete
- Set Permissions

The **Runner** permission role grants all Run capabilities.

The **Editor** permission role grants all Run and Edit capabilities.

Details on flow capabilities and templates
The image above shows a row for each of the permission role templates, illustrating which capabilities they grant or deny.

Capabilities:

- **View** allows a user to view the flow.
- **Run** allows a user to run the flow.
- **Save** allows a user to publish flows and overwrite published flows.
- **Download flow** allows a user to download the flow (as a .tflx).
- **Move** allows a user to move flows between projects. For more information, see Move content.
- **Delete** allows a user to delete the flow.
- **Set Permissions** allows a user to create permission rules for the flow.
Permission roles:

**Runner** allows the user or group to run the flow on the server.

**Editor** allows the user or group to publish and manage flows.

Data Roles

Data Roles have one grouping of capabilities: View/Edit.

View/Edit

- View
- Save
- Move
- Delete
- Set Permissions

The **Interactor** permission role grants the View capability.

The **Editor** permission role grants all capabilities.

Details on data role capabilities and templates
The image above shows a row for each of the permission role templates, illustrating which capabilities they grant or deny.

Capabilities:

- **View** allows a user to view data roles.

- **Save** allows a user to publish data roles and overwrite published data roles.

- **Move** allows a user to move data roles between projects. For more information, see Move content.

- **Delete** allows a user to delete data roles.

- **Set Permissions** allows a user to create permission rules for data roles.

Permission roles:

- **Interactor** allows the user or group to view and apply data roles on the server.
**Editor** allows the user or group to connect to, delete, and set permissions on data roles on the server. They can also publish data roles, and as long as they are the owner of the data role they publish.

**Project permissions**

Projects organize content and setting permissions at the project level can simplify access management. Characteristics of projects that make them useful for managing content include the ability to create nested hierarchies, hiding projects from certain users or groups, authorizing Project Leaders, and locking permissions for straightforward permissions management.

**Note:** How permissions are set at the project level is very important, especially for the Default project. When a new top-level project is created it inherits its default permission rules (for all content types) from the Default project. When a new project is created nested inside another project, the child project inherits its default permission rules from the parent project.

**Project administration**

Projects are containers used to organize and manage access to content. By giving non-administrators privileges to manage projects, certain content administration tasks can be handled at the project level.

**Project Leaders:** Projects can have project leaders, users who have been given the **Project Leader** capability. This capability automatically grants a user their maximum capabilities—depending on their site role—for that project. Project Leaders with site role of Explorer (can publish) and above will have all capabilities. This essentially creates local admins for the project while denying access to site or server settings to users who shouldn’t have it.

**Hierarchy:** Only administrators can create top-level projects. Project owners and project leaders can create nested projects inside their projects. Project owners and leaders have
full administrative access to the project and its content, as well as any nested projects it contains. In a hierarchy, project leaders are implicitly given project leader access to all child content. To remove project leader access, you must do so at the level in the hierarchy where the role was explicitly assigned.

Ownership: A project can have multiple project leaders, but each project has exactly one owner. By default, a project is owned by the user who created it. A project’s owner can be changed (by the existing owner or an administrator, but not a project leader) to any user with a site role of Explorer (can publish) or Creator, or an administrator site role. Project ownership can be changed regardless of whether the project permissions are locked.

Content ownership can be changed by project owners, project leaders, and administrators.

Deleting: Content can only exist inside a project. Only administrators can create and delete top-level projects, but project leaders can create or delete nested projects. Deleting projects also deletes all the content and nested projects they contain. To delete a project without losing its content, move the content to another project first. Deleting projects cannot be undone.

For more information, see Use Projects to Manage Content Access and Add Projects and Move Content Into Them.

Lock project permissions

Whether or not a project’s permissions are enforced for its content is controlled by the Content Permissions in Project setting. This setting can be configured in two ways, either Locked to the project (recommended) or Managed by owner.

- When permissions are locked to the project, the permission rules set on the project are enforced for all content in the project. Effective permissions are enforced.
- When permissions are managed by owner, the permission rules set on the project are applied to all content by default but can be modified during or after publication. Effective permissions are preliminary.

Note: It is strongly recommended to lock project permissions to ensure a consistent and clear permission strategy.
To configure the **Content Permissions in Project** setting:

1. You must be logged into the site as an Administrator, Project Owner, or Project Leader
2. Open the permissions dialog box for a top-level project
3. Click the **Edit Content Permissions** button and select the desired option in the dialog box that opens

![Content Permissions in Project Dialog](image)

New top-level projects inherit all initial permission rules from the Default project, except for Content Permissions, which is set to **Managed by the owner**. This can be changed to **Locked to the project** if desired. Any child project has the same Content Permissions setting as its parent project.

When permissions are locked to the project:

- The top-level project permission rules are applied to all content and nested projects. When locking a project that was previously managed by the owner, any custom permissions on content are overwritten with the permission rules from the project. This cannot be undone once applied. Unlocking the project (converting back to Managed by the owner) leaves the project's permission rules in place but anyone with the correct capabilities can set new permission rules on content.
- Content moved into a locked project acquires the project's permission settings.
Only administrators and project leaders can modify permission. Any changes to the top-level permission rules propagates to all content in the project. Content owners lose the Set Permission capability but retain all other capabilities on their content.

When permissions are managed by the owner:

- The top-level project permission rules are applied by default when content is published into the project or nested projects are created, but permissions can be modified during publication or after the content is created.
- Content moved into the unlocked project retains its permission rules.
- Any user with the Set Permissions capability can modify permission rules for that content.
- Content owners have all capabilities on their content.

Effective permissions

Permission rules establish who is impacted (a user or group) and what capabilities they are granted or denied for content. Permissions are evaluated in a specific order, yielding effective (sometimes called resulting) permissions on a piece of content.

While it seems straightforward to simply set a permission rule and have that be the whole story, whether a user has a capability may be unclear because of membership in multiple groups and the interplay of site roles and ownership with permission rules.

Here are some common examples of why effective permissions—what the user can or cannot do in actuality—might appear different than what a given permission rule states:

- A user might have a capability they are denied in a permission rule because their site role includes it (administrators).
- A user might have a capability they are denied in a permission rule because their user scenario allows it (because they own the content or are a project owner or leader).
- A user might lack a capability they are allowed in a permission rule because their site role does now allow it.
- A user might lack a capability they are allowed in a permission rule because a conflicting group or user rule denied it.
- A user might lack a capability they are allowed in a permission rule at one level of content (such as a workbook) because another level of content denied it (such as a view).
Tip: To help keep things as straightforward as possible, we recommend (1) setting permission rules for groups instead of users, (2) managing permissions locked at the project level instead of setting permissions on individual content, and (3) setting the All User group’s permission rule to None.

A permission rule establishes each capability as **Allowed**, **Denied**, or **Unspecified**. The permissions grid shows the effective permissions for users, which is always either Allowed or Denied. Selecting a group rule above displays all users in that group below in the grid.

![Permission Grid Example](image)

For example, here we have a group called “Site Roles” with a user for each site role. A permission rule gives this group **Editor** capabilities on **Workbooks**. In the rules area above, we see green checks for every capability. However, in the **User Permissions** area below, we see that several site roles are missing capabilities. Hovering over a capability brings up a tooltip that explains the effective permission. Here, Explorer, Unlicensed, and Viewer are missing capabilities that are granted by the permission rule because their site role prohibits those capabilities.

A capability is allowed for a user if and only if:
that capability is within the scope of their site role

AND

they have that capability

• based on a specific user scenario (such as being the content owner or a project leader, or they’re an administrator site role) OR
• they have been allowed the capability as a user OR
• they are both in a group that has been allowed the capability and no rules deny them the capability as a user or member of another group

AND

• there is no conflicting permissions settings at another content level that takes precedence

Any other situation denies the user the capability.

Evaluate permission rules

Permissions in Tableau are restrictive. Unless a capability is granted to a user, they are denied permission. The following logic evaluates if a capability is allowed or denied for an individual:

1. **Site role**: If a site role does not permit a capability, the user is denied. If the user’s site role does permit the capability, then specific user scenarios are evaluated.
   - For example, a Viewer site role cannot connect to a data source. See General capabilities allowed with each site role for more information on what each site role can do.
2. **Specific user scenarios:**
   - If the user is an admin they have all capabilities on all content.
   - If the user is a project owner or project leader, they have all capabilities on all content in their projects.
   - If the user is the content owner, they have all capabilities* on their content.
   - If these scenarios do not apply to the user, then user rules are evaluated.

*Exception: Content owners will not have the **Set Permissions** capability in projects where permissions are locked. Only administrators and project leaders can set permission rules in locked projects.

3. **User rules:** If the user is denied a capability, it is denied. If they are allowed a capability, it is allowed. If a capability is unspecified, then group rules are evaluated.

4. **Group rules:** If the user is in *any* group that is denied a capability, it is denied. If the user is in a group that is allowed a capability (and not in any groups that are denied that capability), it is allowed.
   - That is to say, if a user is a member in two groups, and one is allowed a capability and one is denied the same capability, the denial takes precedence for that user and they are denied.

5. If none of the above conditions apply, the user is denied that capability. In effect, this means that capabilities left as unspecified will result in denied.

A final effective permission of **Allowed** therefore occurs in three circumstances:

- Allowed by site role (Server Administrator, Site Administrator Creator, Site Administrator Explorer)
- Allowed because the user is the content owner, project owner, or project leader
- Allowed by a group or user rule (and not denied by a rule of higher precedence)

**Denied** occurs in three circumstances:

- Denied by site role
- Denied by a rule (and not allowed by a rule of higher precedence)
- Not granted by any rule

Evaluate permissions set at multiple levels

If project content permissions are **managed by owner**, it's possible to configure permission rules in multiple places. There are specific rules that determine what permissions are
If there are nested projects, permissions set at the child level take precedence over permissions set at the parent level.

Changes to permissions at the project level only impact new content, they are not enforced for existing content.

If there are permissions set on content (workbook, data source, or flow) during or after publication, these take precedence over rules set at the project level.

If a workbook does not show sheet tabs, any changes to the workbook-level permissions will not be inherited by the views and any changes to permissions must be done on the view.

Configuring the workbook to show sheet tabs will override existing view-level permissions and sync them with the workbook-level permissions. See Show or Hide Sheet Tabs.
This image shows how capabilities are evaluated through multiple levels of content.

Permission settings for specific scenarios

Certain actions require combinations of permission capabilities and possibly site roles. The following are some common scenarios and their necessary permission configurations

Save and Download/Save As

In the context of permissions, saving is essentially publishing. As such, the Save and Save As capabilities can only be given to users with a site role of Administrator, Creator, or Explorer (can publish). Explorer or Viewer site roles cannot save or save as.

- The Save capability for content can be thought of as both publish and overwrite.
- The Save As capability can be thought of as publishing a copy of that content with any changes made.

It’s important to note that users aren’t able to save or save as a piece of content unless they have the Save capability for at least one project, because all content must be published into a project. Without the Save capability at the project level, the content cannot be published.

In web editing, the Save option in the File menu only appears to the content owner. If a user has the Save capability (allowing them to overwrite the content), they need to File > Save As and name the workbook the exact same name. This prompts a warning that they are about to overwrite the existing content, which they can do. Conversely, a user with only the Save As capability trying to use the same name gets an error stating they don’t have permission to overwrite the existing content.

If a user who is not the content owner saves (overwrites), they become the owner, with all the permissions that entails. The original owner’s access to the content is then determined by their permissions as a user rather than the owner.

**Note:** Download Workbook/Save As is a joint capability for workbooks. Explorers can be given this capability but they are only able to download the workbook, not save as. Giving the Download Workbook/Save As capability to Explorer (can publish),
Creator, or Administrator site roles gives them both the ability to download workbooks and save as.

Web Editing and Web Authoring

Web editing and web authoring refer to the general ability for users to edit or create workbooks directly in the browser. The permission capability is called Web Edit and the site setting is called Web Authoring. This section will refer to any web-based editing or publishing action as web authoring.

To enable this functionality, there are several requirements.

- **Site setting**: Web authoring must be turned on for the entire Tableau site. See Set a Site’s Web Authoring Access and Functions. Without this setting enabled, no users can create new workbooks or edit existing workbooks from the browser, even if they have the web edit capability.

- **User site role**: The user must have the appropriate site role.
  - Viewers can never web edit.
  - Explorers can be given the web edit capability but cannot publish. Essentially, they can use web editing to answer deeper questions based on existing content on the fly, but cannot save (publish) their edits.
  - Explorers (can publish) or Site Administrator Explorers can save (publish), but they can only use data that is already published to the site.
  - Creators, Site Administrator Creators, and Server Administrators can publish and create new data sources.

- **Permission capabilities**: The user must have the necessary permission capabilities based on the desired functionality. See Set Web Edit, Save, and Download Access on Content.

Required Permission Capability Settings

<table>
<thead>
<tr>
<th>Desired functionality</th>
<th>Minimum Site Role</th>
<th>Web Edit</th>
<th>Download/ Save As (workbook)</th>
<th>Save (project)</th>
<th>Connect (data source)</th>
</tr>
</thead>
</table>

Tableau Server on Linux Administrator Guide
Web edit without being able to save

Web author and save new content

Web author and save (overwrite) content

Web author with new data and save new content

Optional indicates this capability is not involved in the desired functionality

Data access for published Tableau data sources

Data sources published to a Tableau site can have native authentication as well as permissions within the Tableau environment.

When the data source is published to the Tableau site, the publisher can choose how to Set Credentials for Accessing Your Published Data which addresses how data source credentials are handled (such as requiring users to log into a database or enter their credentials for Google Sheets). This authentication is controlled by whatever technology holds the data. This can be embedded when the data source is published, or the data source publisher can chose to prompt the user for their credentials to the data source. For more information, see Publish a Data Source.

There are also data source capabilities that allow or deny users the ability to see (View) and connect to the published data source (Connect) in the context of Tableau. These capabilities are set like any other permissions in Tableau.
When a workbook is published that uses a published data source, the author can control how the Tableau authentication will behave for someone consuming the workbook. The author sets the workbook’s access to the published data source, either as Embed password (using the author’s Connect access to the data source) or Prompt users (using the Connect access of the person viewing the workbook), which may require data source authentication as well.

- When the workbook is set to Embed password, anyone who looks at the workbook will see the data based on the author’s access to the data source.
- If the workbook is set to Prompt users, the Tableau-controlled access is checked for the data source. The person consuming the workbook must have the Connect capability for the published data source to see the data. If the published data source is also set to Prompt user, the viewer must also enter their credentials for the data source itself.

<table>
<thead>
<tr>
<th>Workbook authentication to the data source</th>
<th>Data source authentication to the data</th>
<th>How data access is evaluated for someone consuming the workbook</th>
</tr>
</thead>
<tbody>
<tr>
<td>Embed password</td>
<td>Embed password</td>
<td>User sees the data as if they were the workbook author</td>
</tr>
<tr>
<td>Embed password</td>
<td>Prompt user</td>
<td>User sees the data as if they were the workbook author. (The author is prompted for data source authentication, not the user.)</td>
</tr>
<tr>
<td>Prompt user</td>
<td>Embed password</td>
<td>User must have their own Connect capability to the published data source</td>
</tr>
<tr>
<td>Prompt user</td>
<td>Prompt user</td>
<td>User must have their own Connect capability to the published data source and are prompted for their credentials to the underlying data</td>
</tr>
</tbody>
</table>

Note that this applies to consuming a workbook, not web editing. To web edit, the user must have their own Connect capability.
Move content

To move an item, open its Action menu (...) and click **Move**. Select the new project for the item, then click **Move Content**. If Move is unavailable or there are no available destination projects, verify the appropriate conditions are met:

- Administrators can always move content and projects to any location.
- Project leaders and project owners can move content and nested projects among their projects.
  - Note that non-administrators cannot move projects to become top-level projects.
- Other users can move content only if all three of the following requirements are met:
  - Creator or Explorer (Can Publish) site role.
  - Publishing rights (**View** and **Save** capabilities) for the destination project.
  - Owner of the content, or—for workbooks and flows—having the **Move** capability.

When a project or content is moved, permissions might change. Project leaders or project owners always gain permissions for items moved into their projects.

- When items are moved into a **locked** project, the permission templates for the locked project are enforced on the moved item. Note that this might strip the user moving the item of their ability to move it again if they don’t have the correct permissions in the locked project.
- When items are moved into an **unlocked** project (managed by the owner), the existing permissions are retained for the moved item. If the project leader capability on the moved item has only implicitly been granted (from a higher-level project), that capability is removed, though any explicitly set project leader capabilities is retained.

Show or Hide Sheet Tabs

Although it is not recommended as a general practice, there are times when it can be useful to set permissions on views independently of the workbook that contains them. To do so, two conditions must be met. (1) The workbook must be published into an unlocked project and (2) the workbook cannot show sheets as tabs.

When a workbook shows sheets as tabs, all views inherit the workbook permissions and any changes to the workbook permissions affect all of its tabbed views. When a workbook is
published without showing sheets as tabs, all views assume the workbook permissions upon publication, but any subsequent changes to the workbook’s permission rules will not be inherited by the views. View-level permissions can be set only on views that are already published, not during the publishing process.

Changing the configuration of sheets as tabs on a published workbook will also impact the permission model. Show Tabs will override any existing view-level permissions and reinstate the workbook-level permissions for all views. Hide Tabs will break the relationship between the workbook and its views.

- To configure sheets as tabs on a published workbook, open the Actions menu (...) for the workbook and select Tabbed Views. Choose Show Tabs or Hide Tabs as desired.
- To configure sheets as tabs during publishing, refer to Show sheets as tabs.
- To set view-level permissions, see Set permissions on content.

Remember, in an unlocked project, any modifications to the workbook-level permissions will not be applied if sheet tabs are hidden.

**Quick Start: Permissions**

You can use permission rules to control access to content on a site. A permission rule is a set capabilities that defines the level of access a group or user has on a content item. Content items are projects you create, and the workbooks and data sources published to them.

The most efficient way to manage permissions is to remove permissions from the All Users group before you create new groups for your environment. Then assign the permissions to groups at the project level.

1 Add users to groups

Create groups for users based on who should have the same permissions, and then add users to those groups. Within a site, select Groups. Select a group name, and then select Add Users.
2 Open a project’s permissions settings

The site’s Content page shows the top-level projects. Navigate to the project you want to update, select its Actions (…) menu, and then select Permissions.
If you're not sure where to find a child project, display filters, and select **Show all projects**.

3 Create a permissions rule

Click **Add a user or group rule**, select **Group**, and then find and select the group.

Select a permissions role template to apply an initial set of capabilities for the group. Click a capability to set it to **Allowed** or **Denied**, or leave it **Unspecified**. Click **Save** when you are done.
Whether a user can set permissions is based on their site role and how their **Set Permissions** capability is set.

4 View a user's effective permissions

After you save the permissions rule for the group, you can view the effective permissions for that content.

Click a group name to see the group’s users and their permissions. Hover over a capability box to see a tooltip with details on whether a capability is allowed or denied.
Custom indicates a user’s capabilities have been changed from the initial settings for their site role or content role.

For more information, see Permissions.

Site roles

A user’s site role determines the maximum permissions allowed for that user.

- Server and site administrators can access all site content and take actions on it.
- Owners always get full access to the content they’ve published. When the parent project permissions are not locked, owners can change permissions for their published content.

For more information, see Set Users’ Site Roles and Use Projects to Manage Content Access.

Permissions evaluation

- **Denied** takes precedence over **Allowed**.
- **Unspecified** results in **Denied** if no other permissions are specified.
Specific user permissions on content take precedence over group permissions on content. In other words, user permissions trump group permissions.

For a couple of best-practice steps for how to implement permissions, see the following:

- Configure Projects, Groups, and Permissions for Managed Self-Service
- Structure Content Projects, Groups, and Permissions (links to Everybody’s Admin Guide)
Manage Content Ownership

When you publish a data source or workbook on Tableau Server or when you create a project, you become its owner. A content owner, a project leader with an appropriate site role, or an administrator can change ownership of a content asset. After ownership is reassigned, the original owner has no special connection to the content item, and their ability to access it is determined by their permissions on the project or that specific item.

Who can change or be given ownership, by content type

Whether you can change or be given ownership depends on your permissions and your relationship to the content asset, as described in the following table.

Note: Full project leader access is available only with some site roles. For information, see Project-level administration.

<table>
<thead>
<tr>
<th>Content asset type</th>
<th>Who can change ownership</th>
<th>Who can be given ownership</th>
</tr>
</thead>
<tbody>
<tr>
<td>Top-level projects</td>
<td>Server administrator¹</td>
<td>Server administrator</td>
</tr>
<tr>
<td></td>
<td>Site administrator</td>
<td>Site administrator (Creator and Explorer)</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Creator</td>
</tr>
<tr>
<td>Child (nested) projects</td>
<td>Server administrator</td>
<td>Any administrator or user of the site, excluding users with a Read Only site role.</td>
</tr>
<tr>
<td></td>
<td>Site administrator</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Project leader or owner</td>
<td></td>
</tr>
<tr>
<td>Workbooks and data sources</td>
<td>Server administrator</td>
<td>Any administrator or user of the site, excluding users with a Read Only site role.</td>
</tr>
<tr>
<td></td>
<td>Site administrator</td>
<td></td>
</tr>
<tr>
<td>Workbook or data source owner</td>
<td>Project leader or owner of the project that contains the item</td>
<td></td>
</tr>
<tr>
<td>--------------------------------</td>
<td>---------------------------------------------------------------</td>
<td></td>
</tr>
<tr>
<td><strong>Flows</strong></td>
<td><strong>Server administrator</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Site administrator</strong></td>
<td></td>
</tr>
<tr>
<td></td>
<td><strong>Any administrator or user of the site, excluding users with a Read Only site role.</strong></td>
<td></td>
</tr>
</tbody>
</table>

1 The Server Administrator site role applies to Tableau Server only; not Tableau Online.

Considerations for changing content ownership

- Before you remove a user from Tableau Server, make sure they do not own any content assets.

  If the user does own content, you must first reassign ownership of those assets before you can delete the user. Otherwise, their site role is set to **Unlicensed**, but they are not deleted, and only an administrator can take certain actions on that content.

- If you change the ownership of a flow, workbook or data source that includes embedded credentials, to connect to underlying data, the embedded credentials will be deleted.

  You can update the embedded credentials by editing the connection information on Tableau Server. For more information, see **Edit Connections**. Alternatively, the new owner can download the flow, workbook or data source, and open the item in
Tableau Server on Linux Administrator Guide

Tableau Desktop to update the embedded credentials, and then re-publish the workbook or data source.

- If you do not lock permissions to projects, make sure users you give content ownership to know your permissions guidelines, or you account for permissions as you change ownership. In unlocked projects, by default, content owners can set permissions on their content. For more information, see Lock project permissions.

Change the owner of a content resource

1. Sign in to the Tableau Server web environment and select the Content tab.

2. Navigate to the content you want to assign to someone else.

   - If you want to re-assign multiple of the same type of content, for example, multiple workbooks, open the Explore menu, and select the content type.
   
   - If you want to re-assign multiple items within the same project, navigate to the project.

   If you’re not sure where to find a child project, display filters, and select Show all projects.

3. Select the items you want to reassign, and then select Actions > Change Owner.

   The other menu commands you see will depend on the content type.
4. Type the name of a user or select a user from the list.

![Change owner dialog]

5. Click **Change Owner**.

**Manage Permissions for External Assets**

Tableau Online and Tableau Server provide a space for accessing and managing published content. When Tableau Online or Tableau Server is licensed with the Data Management...
Add-on, you have access to Tableau Catalog. Tableau Catalog adds a complementary space and a set of features across your site to track and manage metadata and lineage of external assets used by the content published to your site.

Tableau Catalog indexes content and assets

Catalog discovers, tracks, and stores metadata from the content that you publish to Tableau Online or Tableau Server.

Catalog indexes metadata for the following:

- **Tableau content**: workbooks, data sources, flows, projects, users, and sites

- **External assets**: databases and tables associated with Tableau content

  Catalog classifies the metadata of any data that comes from outside the Tableau environment as external assets. The data that comes from outside the Tableau environment is stored in many different formats, such as a database server or a local .json file.

  Catalog tracks only the metadata of the external data and does not track the underlying data in any form (raw or aggregated).

Catalog metadata includes the following:

- **Lineage information** or the relationship between items. For example, the Sales table has a relationship with both the Superstore data source and the Superstore Sample workbook.

- **Schema information**. Some examples include:
  - Table names, column names, and column types. For example, Table A contains Columns A, B, and C, which are types INT, VARCHAR, and VARCHAR.
  - Database name and server location. For example, Database_1 is a SQL Server database at http://example.net.
  - Data source name, and the names and types of the fields the data source contains. For example, Superstore data source has fields AA, BB, and CC. Field CC is a calculated field that refers back to both field AA and field BB.
- **User curated, added, or managed information.** For example, item descriptions, certifications, user contacts, data quality warnings, and more.

How does Tableau Catalog work?

Tableau Catalog indexes all content published to Tableau Online or Tableau Server to track lineage and schema metadata. For example, the metadata comes from workbooks, packaged workbooks, data sources, and the Tableau Server or Tableau Online repository.

As part of the indexing process, lineage and schema metadata about external assets (databases and tables) used by the published content are also indexed.

**Note:** In addition to accessing Catalog from Tableau Online or Tableau Server, indexed metadata can also be accessed from the Tableau Metadata API and Tableau Server REST API. For more information about the Tableau Metadata API or metadata methods in the REST API, see **Tableau Metadata API** and **Metadata Methods** in the Tableau Server REST API, respectively.

Permissions on assets and their metadata

Permissions control who is allowed to see and manage external assets and what metadata is shown through lineage.

**Note:** If Tableau Online or Tableau Server is not licensed with the Data Management Add-on, then, by default, only Tableau Server and site admins can see database and table metadata through the Tableau Metadata API. This default can be changed to use "derived permissions," as described below.

Access metadata about content and assets

The permissions used to access metadata through Catalog work similarly to permissions for accessing content through Tableau Online or Tableau Server, with some additional considerations for external assets.
Permissions on Tableau content

Catalog uses the "view" permissions that are already used by existing Tableau content to control the metadata that you can see on Tableau content. For more information on "view" permissions on content, see Effective permissions.

Permissions on external assets using derived permissions

When Tableau Online or Tableau Server is licensed with the Data Management Add-on, Catalog uses derived permissions to automatically grant you "view" permissions to external assets in the following scenarios:

- If you are the owner of a workbook, data source, or flow, you can see the database and table metadata used by that workbook, data source, or flow.

- If you are a project owner or project leader, you can see all the database and table metadata used by the content published to your project.

Check permissions

As a Tableau Server admin, Tableau Online site admin, or someone who has been given the capability to set permissions for an asset, you can validate who has derived permissions by following the steps below.

1. Sign in to Tableau Online or Tableau Server.

2. From the left navigation pane, click External Assets.

3. From the drop-down menu, select Databases and Files or Tables.
   
   Note: Local files, like .json or .csv files are grouped as external assets under Databases.

4. Select the check box next to the database or table whose permissions you want to modify, and then select Actions > Permissions.

5. In the Permissions dialog box, click Add a user or group rule, select Group or
User, and then select the group or user name from the list.

6. Validate the permissions by doing the following:
Click a group name or user name in the permission rules to see the results. Hover over a capability box to see a tooltip that shows whether a capability is allowed or denied, and what determined that result.

Order of precedence in which Tableau evaluates derived permissions for external assets

When derived permissions are configured for your Tableau Online site or Tableau Server, each user’s level of access to external assets depends on the associated Tableau content and the order of precedence of rules Tableau uses for its content.

Tableau follows the rules below, continuing on to the next rule, only if the current rule evaluates to "denied." If any rule evaluates to "allowed," the capability is allowed and Tableau stops evaluating. This rules list is based on the Evaluate permission rules.

1. Admin role
2. License
3. Project leader (Tableau content)
4. Project owner (Tableau content)
5. Content owner (Tableau content)
6. Derived permissions (applies only to external assets and the "view" capability)
   a. Admin role
   b. License
   c. Project leader (external assets)
   d. Project owner (external assets)
   e. Content owner (external assets)
7. Explicit permissions

Turn off derived permissions

As Tableau Online site admin or Tableau Server admin, you can turn off the derived permissions default setting for a site in favor of manually granting explicit permissions to databases and tables.

1. Sign in to Tableau Online as a site admin or Tableau Server as a Server admin.
2. From the left navigation pane, click Settings.
3. On the General tab, under Automatic Access to Metadata about Databases and Tables, clear the Automatically grant authorized users access to metadata about databases and tables check box.

Note: Data quality warning messages on databases and tables that are visible to users though derived permissions remain visible to those users even when the check box is not selected.

Set permissions on individual external assets

In order to grant additional users permissions to view, edit, and manage external assets, a Tableau Server or site admin can grant those capabilities explicitly on individual databases or tables for users or groups.

Database permissions act as a permissions template

Database permissions function like Project permissions. In other words, when permissions are set at the database level, those permissions can serve as a template for any newly
discovered and indexed child tables of that database. Furthermore, database permissions can also be locked so that the child tables will always use the permissions set at the database level.

Granting permission at the database level can help create a scalable process for enabling permissions to tables.

**Summary of permissions capabilities**

The following table shows the capabilities you can set for external assets (databases and tables):

<table>
<thead>
<tr>
<th>Capability</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>🎦 View</td>
<td>See the database or table asset.</td>
</tr>
<tr>
<td>📝 Save</td>
<td>Add or edit data quality warnings and descriptions of the database or table asset.</td>
</tr>
<tr>
<td>📫 Set Permissions</td>
<td>Grant or deny permissions for the database or table asset.</td>
</tr>
</tbody>
</table>

**Set permissions on a database or table**
Tableau Server on Linux Administrator Guide

To set permissions on databases or tables, use the following procedure.

1. Sign in to Tableau Online or Tableau Server as an admin or someone who has been granted the "Set Permissions" capability.

2. From the left navigation pane, click **External Assets**.

3. From the drop-down menu, select **Databases and Files** or **Tables**.
   
   **Note**: Local files, like .json or .csv files are grouped as external assets under **Databases**.

4. Select the check box next to the database or table whose permissions you want to modify, and then select **Actions > Permissions**.

5. In the Permissions dialog box, click **Add a user or group rule**, select **Group or User**, and then select the group or user name from the list.

6. Select a permission role template to apply an initial set of capability for the group or user, and then click **Save**. Available templates are: Viewer, Editor, None, and Denied.

7. To further customize the rule, click the actions menu (...) next to the rule name, and then click **Edit**. Click a capability in the rule to set it to Allowed or Denied, or leave it unspecified. Click save when you are done.

8. Configure any additional rules you want for other groups or users.

9. Validate the permissions by doing the following:
   
   Click a group name or user name in the permission rules to see the results. Hover over a capability box to see a tooltip that shows whether a capability is allowed or denied,
and what determined that result.

Lock permissions to the database

To lock (or unlock) permissions to the database, use the following procedure.

1. Sign in to Tableau Online or Tableau Server as an admin or someone who has been granted the "Set Permissions" capability.

2. From the left navigation pane, click External Assets. By default, the External Assets page shows a list of databases and files.

3. Select the check box next to the database whose permissions you want to lock, select Actions > Permissions, and then click the Edit Content Permissions button.

4. In the Table Permissions in Database dialog box, select Locked to the database and click Save.

5. To unlock permissions, click Edit Content Permissions again, and select Managed by the owner.
Access lineage information

Catalog (and the Metadata API) can expose relationship and dependencies metadata, also referred to as lineage, among the content and assets on Tableau Online or Tableau Server. Lineage can show three primary things:

- How items relate to each other, either directly or indirectly
- How many of those items relate to each other
- With the appropriate permissions, shows sensitive data about items in the lineage, such as data quality warning messages

**Sensitive lineage data**

In some cases, lineage can contain sensitive data, such as content or asset names.

By default, complete lineage information displays for all users while its sensitive data is blocked from specific users who don’t have the appropriate “view” permissions. The concept of blocking sensitive data is called obfuscation.

Obfuscation allows all metadata in the lineage to be visible while keeping its sensitive data blocked from specific users who don’t have the appropriate “view” permissions. This default enables workflows that rely on a complete impact analysis.

If obfuscating sensitive data in the lineage is not enough for your organization, certain parts of the lineage, including its sensitive data, can be filtered.

Filtering omits certain parts of the lineage for specific users who don't have the appropriate "view" permissions to its sensitive data. Because filtering omits parts of lineage, it prevents workflows that rely on a complete impact analysis.

To change how sensitive data in a lineage is handled, do the following:

1. Sign in to Tableau Online as a site admin or Tableau Server as a Server admin.
2. From the left navigation pane, click Settings.
3. On the General tab, under Sensitive Lineage Information, select the radio button that can best handle lineage information for all users on your Tableau Online site or Tableau Server.

**Additional notes about lineage**
• **If you don't have "view" permissions on related assets**, you can always see when assets relate to each other.

For example, you can see 1) whether related upstream databases and tables exist in the lineage and 2) the total number of databases or total number of tables that are related to the asset you are evaluating.

However, you can't see the metadata associated with those assets when you don't have view permissions for them. When metadata is blocked because of limited permissions, you see **Permissions Required**.

• **If you don't have "view" permissions on related assets**, you can always see whether the assets are certified.

However, the level of detail that you can't see if you don't have "view" permissions is the sensitive information related to the certification, like the names of the related
databases and tables. When metadata is blocked because of limited permissions, you see Permissions Required.

<table>
<thead>
<tr>
<th>Tables (6)</th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Type</td>
<td>Name</td>
<td>Workbooks</td>
<td>Data Sources</td>
</tr>
<tr>
<td></td>
<td>Permissions Required</td>
<td></td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>Permissions Required</td>
<td></td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>Permissions Required</td>
<td></td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>REI</td>
<td></td>
<td>2</td>
<td>4</td>
</tr>
<tr>
<td></td>
<td>Permissions Required</td>
<td></td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>0</td>
<td>5</td>
</tr>
</tbody>
</table>

- **If you have "view" permissions on related assets**, you can see when and what assets and content are related to each other, and their sensitive metadata.

For example, you can see 1) the names, data quality warnings, and total number of related upstream databases and tables and 2) the combined number of sheets (visible and hidden) in the lineage of the downstream workbook of the asset you are evaluating.

For more information about lineage see Use Lineage for Impact Analysis.

**Potential mismatch between asset results and content results**

When Catalog shows lineage information, it provides information between content and assets. Catalog lineage always shows the true count or result of associated items. However, elsewhere in Tableau Online or Tableau Server, you might see fewer number of items. One reason for this is because of your "view" permissions. Outside of Catalog, or elsewhere in Tableau Online or Tableau Server, you see a filtered count or result of the content that you have access to according to your content permissions.
For example, suppose you’re looking at the Superstore data source. The lineage for the Superstore data source can show how many upstream underlying tables the data source connects to and how many downstream workbooks rely on the data source. However, because you might not have "view" permissions on all of those downstream workbooks, the total number of related workbooks might be different when you’re looking at Catalog lineage information versus the total number of workbooks represented in the Connected Workbooks tab.

There might be other reasons why, which are not related to permissions, you might see a mismatch between asset counts and content counts. For more information, see Use Lineage for Impact Analysis.

Manage Data

You can connect to and manage the data you that you use in Tableau.

Tableau Server Data Sources

When your Tableau users want to share data connections they’ve defined, they can publish data sources to Tableau Server. When a data source is published to the server, other users can connect to it from their own workbooks, as they do other types of data. When the data in the Tableau data source is updated, all workbooks that connect to it pick up the changes.

Looking for Tableau Server on Windows? See Tableau Server Data Sources.

A Tableau Server data source consists of metadata that describes the following:

- **The connection information**: Defines whether the data is in a live database or an extract, and which of that data to bring in to Tableau.

- **Customization and cleanup**: Includes information that facilitates efficient use of the data. For example, calculations, sets, groups, bins, parameters, custom field
Data access and refresh instructions: Includes the location of the underlying database server (whether on-premises or in the cloud), network paths for file-based data, security information such as credentials or access tokens, and related information.

In addition to helping your users create data consistency and reliability, using Tableau data sources offers advantages to you as the administrator. Because multiple workbooks can connect to one data source, you can minimize data source proliferation and save on storage space and processing time. When someone downloads a workbook that connects to a Tableau data source that in turn has an extract connection, the extract stays on the server, reducing network traffic. Finally, if a connection requires a database driver, you need to install and maintain the driver only on the server, instead of on each user’s computer. If you use Tableau Online, all supported drivers are available to data sources published to your site.

Managing data sources

You can perform some or all management tasks on a data source if you have one of the following levels of access:

- Site or server administrator
- Project leader or owner of the project the data source is published to
  
  Full project leader access is available only with some site roles. For information, see Project-level administration.

- Data source owner

Unless you’re a site or server administrator, you might not have access to all management tasks. For example, if the Permissions menu is not available, chances are that the project permissions are locked, which denies setting permissions on individual workbooks and data sources.
**Tip:** A best practice is to designate a person or team to manage all data sources published to a project or to the whole site, so that all data sources can be maintained under the same set of guidelines.

To perform the management tasks that you have access to, do the following:

1. Sign in to the site, and on the **Content** tab, select **Explore > Data sources**.

2. On a data source, select the **Actions (…) menu**.

   ![Actions menu](image)

   - **New workbook or Download:** Create a new workbook in the browser environment that connects to this data source. Or download the data source to use locally.

   - **Tag:** Add or remove keyword tags. Tags can contain a single word or multiple words, delimited by a comma.

   - **Move:** Move a data source from one project to another. This requires specific settings on each project. For information, see Required access level for moving content.

   - **Permissions:** View or update permissions that specify which users or groups can
connect to, modify, or download the data source. As mentioned at the beginning of this section, if this action is not available, the project permissions might be locked, and only the project leader or administrator can change permissions.

- **Change owner**: Making someone an owner gives them complete access to it.

- **Refresh extracts**: If a data source includes an extract, you can assign the extract to a refresh schedule.

  For information, see Refresh Data on a Schedule.

- **View the data source’s revision history**

- **Delete**: Deleting a data source affects workbooks that connect to the data source. Before you delete a data source, ensure that there are no workbooks that connect to the data source or edit the workbooks to use another data source.

In addition, for data sources that are proxy connections, administrators can stay aware of how users authenticate to the database, and whether the appropriate drivers are installed. For information, see Database Drivers and Data Security.

### Extract Upgrade to .hyper Format

Beginning in version 10.5, new extracts use the .hyper format instead of the .tde format. Extracts in the .hyper format take advantage of an improved data engine, which supports the same fast analytical and query performance as the data engine before it, but for even larger extracts.

This article covers what you and your users should be aware of as a result of the extract format change in version 2019.4 and how the change impacts your users.

### Changes to extracts in version 10.5

With the introduction of Tableau's improved data engine technology, Hyper, comes a change to the extract format. As mentioned above, extracts created in version 2019.4 use the .hyper format. Though Tableau version 2019.4 can continue to read .tde extracts, it cannot create
new .tde extracts. The impact of this format change means that when certain extract tasks are performed either by a user or by Tableau Server, such as an extract refresh or append data, the .tde extract is automatically upgraded and converted to a .hyper extract.

After an extract is upgraded to a .hyper extract, your users can interact with the .hyper extract as they would a .tde extract. However, your users should be aware of the following backward limitations:

- The upgrade can't be reversed. An upgraded extract can't be converted back to a .tde extract.
- The upgraded extract can't be opened in Tableau Desktop 10.4 and earlier.
- Workbooks that contain upgraded extracts can't be published as an older version from Tableau Desktop to Tableau Server.
- Workbooks that contain upgraded extracts can't be downgraded as an older version in Tableau Desktop.
- Workbooks that contain an upgraded extract, which are downloaded from Tableau Server using the Download Tableau Workbook option, cannot be opened in Tableau Desktop 10.4 and earlier.

Tasks that cause extract upgrade

There are three distinct ways a .tde extract can get upgraded to a .hyper extract: 1.) during an extract refresh (full or incremental), 2.) when appending data to an extract, and 3.) when an extract is upgraded manually using Tableau Desktop 2019.4. After an extract has been upgraded, the original .tde extract is automatically removed from Tableau Server if it's not being reference by other workbooks.

The followings tasks on Tableau Server automatically upgrades and converts a .tde extract to a .hyper extract:

- Manual extract refresh
Tableau Server on Linux Administrator Guide

- Scheduled extract refresh
- Automated refresh task that's performed through tabcmd
- Automated refresh task that's performed through the Extract Command-Line Utility
- Automated append data to an extract using tabcmd
- Automated append data to an extract using the Extract Command-Line Utility
- Encryption of a .tde extract. For more information, see Extract Encryption at Rest.

Impact of extract upgrade

Tableau recommends that the Tableau Desktop version in your environment be upgraded to match Tableau Server 2019.4. If they don’t match, extract compatibility can be an issue depending on the task your users want to perform.

To get a better understanding of when extract upgrades can occur and potential compatibility issues your users might experience, review a detailed explanation of the extract compatibility scenarios on the Tableau Support page or a summary explanation below.

**Note:** In the tables below, "10.4" represents Tableau 10.4 and earlier and "10.5" represents Tableau 10.5 and later.

Performing tasks on Tableau Server 10.5

When working with extracts created in Tableau Desktop 10.4 and earlier, your users should be aware of the following extract-related compatibility scenarios around common tasks performed on Tableau Server 10.5.

<table>
<thead>
<tr>
<th>Task</th>
<th>10.4 workbook</th>
<th>10.5 workbook</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>.tde extract</td>
<td>.tde extract</td>
</tr>
<tr>
<td>Publish from Tableau</td>
<td>✓</td>
<td>Not possible</td>
</tr>
</tbody>
</table>
## Downloading from Tableau Server 10.5

When working with extracts created in Tableau Desktop 10.4 and earlier, your users should be aware of the following extract-related compatibility scenarios when downloading from Tableau Server 10.5.

<table>
<thead>
<tr>
<th>Task</th>
<th>10.4 workbook</th>
<th>10.5 workbook</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>.tde extract</td>
<td>.tde extract</td>
</tr>
<tr>
<td>Desktop 10.4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Publish from Tableau Desktop 10.5</td>
<td>√</td>
<td>√</td>
</tr>
<tr>
<td>Refresh, scheduled refresh, or append</td>
<td>Workbook version remains unchanged, but extract upgrades to .hyper format</td>
<td>Workbook version remains unchanged, but extract upgrades to .hyper format</td>
</tr>
<tr>
<td>Edit/save in web authoring</td>
<td>Workbook version changes to 10.5, extract remains in .tde format</td>
<td>√</td>
</tr>
<tr>
<td>Edit/save in web authoring then refresh or append</td>
<td>Workbook version changes to 10.5, and extract upgrades to .hyper format</td>
<td>Workbook version remains unchanged, but extract upgrades to .hyper format</td>
</tr>
</tbody>
</table>
## Automating refresh and append tasks

When working with extracts created in Tableau Desktop 10.4 and earlier, your users should be aware of the following extract-related compatibility scenarios around automating refresh and append tasks using tabcmd or the Tableau Command-Line Utility.

<table>
<thead>
<tr>
<th>Task</th>
<th>10.4 workbook</th>
<th>10.5 workbook</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>.tde extract</td>
<td>.tde extract</td>
</tr>
<tr>
<td>Download and open</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>in Tableau Desktop 10.4</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Download and open</td>
<td>✓</td>
<td>✓ ✓</td>
</tr>
<tr>
<td>in Tableau Desktop 10.5</td>
<td></td>
<td>✓ ✓</td>
</tr>
<tr>
<td>Export As Version</td>
<td>✓</td>
<td>✓ ✓</td>
</tr>
<tr>
<td>from Tableau Desktop 10.5</td>
<td></td>
<td>✓ ✓</td>
</tr>
<tr>
<td>Task</td>
<td>10.4 workbook .tde extract</td>
<td>10.5 workbook .tde extract</td>
</tr>
<tr>
<td>------</td>
<td>-----------------------------</td>
<td>-----------------------------</td>
</tr>
<tr>
<td>Refresh or append</td>
<td>using 10.4 tabcmd</td>
<td>Workbook version remains unchanged, but extract upgrades to .hyper format</td>
</tr>
<tr>
<td></td>
<td>using 10.5 tabcmd</td>
<td>Workbook version remains unchanged, but extract upgrades to .hyper format</td>
</tr>
<tr>
<td>Publish</td>
<td>using 10.4 tabcmd</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>using 10.5 tabcmd</td>
<td>Not possible</td>
</tr>
<tr>
<td>Refresh or append</td>
<td>using 10.4 Tableau Command-Line Utility</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>using 10.5 Tableau Command-Line Utility</td>
<td>Workbook version remains unchanged, but extract upgrades to .hyper format</td>
</tr>
</tbody>
</table>

Why keep an extract in .tde format?

If the version of Tableau Desktop cannot be upgraded to match Tableau Server 2019.4, your users will need to keep their extracts in .tde format.
How to keep an extract in .tde format

To keep an extract in the .tde format, the extract should not be upgraded. To help your users keep their extracts from upgrading, advise them against performing any of the tasks listed above in the Extract Upgrade to .hyper Format section. Then, consider the following suggestions to maintain a .tde version of an extract:

- Disable existing extract refresh schedules on Tableau Server until you can identify which extracts should and shouldn't be upgraded.

  **Note:** To access an extract's refresh schedule, you must be a data source owner, or be a server administrator, site administrator, or Project Leader. For more information about which site roles allow full Project Leader capabilities, see Project-level administration.

- Continue to perform extract refresh and append data tasks using Tableau Server 2019.4, but maintain an earlier version of Tableau Desktop and the ability to connect to the original data in case you need to recreate the .tde extract.

- Use an earlier version of Tableau Desktop to perform extract tasks, such as extract refresh or append data.

**What to expect after extract upgrade**

After an extract upgrade has taken place, your users should expect some additional changes when working with extracts in version 2019.4. For more information, see **After an extract upgrade** section in Tableau Help.

**Create Extracts on the Web**

You can create extracts in the browser, without using Tableau Desktop.

**Limitations**

- Your connection credentials must be embedded in the data source.
You can't create extracts for embedded data sources that reference published data sources. As a workaround, create the extract directly on the published data source.

In the web, you can't specify extract settings like incremental refresh and extract filters.

This feature does not apply to file-based data sources. File-based data sources already have special performance features and adding extraction will have no performance benefit.

This feature does not apply to bridge-based data sources in Tableau Online.

Extract a Published Data Source

To extract a published data source:

- On the Content tab, select Explore > Data sources.
- Select a data source by clicking on the Data Source name.
- At the top of the screen, under the Data Source name, select the drop-down menu that says Live.
- Change the connection type from Live to Extract. If the extract encryption at rest feature is enabled on the site, select either Encrypted or Unencrypted.
- If you see an error message about embedded credentials, embed your credentials in the data source. To do this, click Edit Connection. Select "Embedded password in connection" and then click Save.

Extract a Data Source that is Embedded in a Published Workbook

To extract the data source:

- Navigate to the published workbook.
- Navigate to the Data Sources tab
- Select one or more of the data sources.
- Click the Action button.
- Click Extract. If the extract encryption at rest feature is enabled on the site, select either Encrypted or Unencrypted.
View Data Source Attributes

In the **Content** area of the Tableau Server web authoring environment, you can filter the view to show only data sources or connections and their attributes.

**View data sources by name**

To filter by data source name, under **Explore**, select **Data sources**.

Attributes in the data source name list include the following:

- **Type**—The icon next to the data source name indicates whether the data source is published as a live connection (.), an extract connection (.), or is embedded in a workbook (.)
  
  - Select the name of a published data source to open its data source page, with tabs for viewing connections and workbooks that connect to it.
  
  - Select the name of an embedded data source to open the workbook associated with it, as well as tabs for viewing other data it connects to.

- **Connects To**—Indicates the name of the database server or data file. This could be a database outside of Tableau Server, an extract, or a published data source.
• **Live or Last Extract**—This column tells you whether the connection to the data is live, or, if it is a connection to an extract, when the extract was last updated.

**View a list of connections**

To filter by connection type, in the View list, select **Connections**.

Connection attributes include:

• **Connects to**—Indicates the name of the database server or data file. This could be a database outside of Tableau Server, an extract, or a published data source.

• **Connection type**—Shows the type of data. **Tableau Server** indicates that the connection is to a data source published on the site. **Tableau Data Engine** means the data source has an extract stored on the Tableau data server.

**Use Certification to Help Users Find Trusted Data**

In a self-service environment with multiple publishers, it's common for a project on Tableau Server to contain a variety of content that is named similarly, or is based on the same or similar underlying data, or is published without any descriptive information about it. When this is the case, analysts might lack confidence about the data they should use.

To help your users find the data that's trusted and recommended for their type of analysis, you can **certify** the data that complies with your organization's data standards.

Certification complements the Recommendations Training Schedule feature by offering a way to promote data through curation.
Starting in 2019.3, Tableau Catalog is available in the Data Management Add-on to Tableau Server and Tableau Online. When Tableau Catalog is enabled in your environment, in addition to certifying published data sources, you can also certify the databases and tables that are associated with your workbook, flow, or data source content. For more information about Tableau Catalog, see "About Tableau Catalog" in the Tableau Server or Tableau Online Help.

How certification helps users find trusted data

When you certify a data source (or database or table, if you have Tableau Catalog in your environment), it gets a green check mark on its icon in the list view or thumbnail in the grid view. This appears anywhere the data source (or database or table) is shown, including when connecting to data in web authoring or from Tableau Desktop.
Certified data sources rank higher in search results and are added to recommended data sources.

In addition, you can provide notes about the certification status, which appear in a tooltip when users hover over the data source icon or card. The tooltip in list view also shows who certified the data source and the data source's connection type.

For more information, see the How to certify data steps below.

Create guidelines for selecting data to certify

As with most Tableau functionality, certification is flexible. You can define for your organization the criteria you use to determine when to certify a data source (or database or table). As you do this, you might want to document and share your guidelines. As new data sources are published, the guidelines can help you and other administrators or project leaders to be consistent with your certification choices. They can also help users understand what certification means.

Whether you use the same certification criteria across all projects, or define unique criteria for each project, the important thing is to be clear about what certification means in your environment.

Who can certify data

To certify a data source, you must
Tableau Server on Linux Administrator Guide

- be an administrator, or
- have a site role of Explorer (Can Publish) or Creator and have the Project Leader capability on the project containing the data you want to certify.

To certify databases or tables, you must have Tableau Catalog enabled in your environment and either of the following permissions levels:

- Site Administrator site role.
- "Manage permissions" capability on a database to certify that database or any tables within that database.

How to certify data

The data you can certify depends on whether you have Tableau Catalog enabled in your environment. All users with the right permissions can certify data sources. Tableau Catalog users with the right permissions can also certify databases, tables, and files.

Certify data sources

1. Sign in to Tableau Server.
2. To certify a data source, on the Explore page, select Data Sources.
3. On the page, select the data source, and then select the Details icon.
4. Next to Certification, select Edit Certification Status, and then do the following:
   a. Select the This data is certified check box.
   b. Add a note that gives users context for the certification status, intended use for the data, or other helpful information.

Information you add to the Note section appears in the certification badge tooltip, mentioned earlier in How certification helps users find trusted data.
Certify databases and tables

When Tableau Catalog is enabled in your environment, you can certify databases and tables, if you have the right permissions.

1. Sign in to Tableau Server.
2. To certify a database or table, on the External Assets page, select Databases and Files or Tables.
3. On the page, select the More actions menu (…) next to the asset name you want to certify.
4. Select Edit Certification and then do the following:
   a. Select the This data is certified check box.
   b. Add a note that gives users context for the certification status, intended use for the data, or other helpful information, and then click Save.

Information you add to the Note section appears in the certification badge tooltip, mentioned earlier in How certification helps users find trusted data.
Keep Data Fresh

You can manually refresh data, as well as schedule data refreshes.

Manage Refresh Tasks

Administrators can change the priority of scheduled extract refreshes relative to other server tasks, manually refresh extracts, or delete their schedules.

1. Sign in to the site that has the schedules you want to manage, and then click **Tasks**.

2. Select one or more scheduled extract refreshes.

3. From the **Actions** menu, do any of the following:

   - Select **Change Schedule**, and choose a new schedule from the list.

   - Select **Run Now** to refresh manually.

   **Note:** If an extract does not have a scheduled refresh, you can refresh it on demand from the Data Connections page.

   - Select **Change Priority**, and enter a number between 1 and 100 to move the extract up or down in the priority list.

   - Select **Delete** to completely remove the schedule for the selected data sources.
Refresh Data on a Schedule

You can schedule refresh tasks for published extract data sources and published workbooks that connect to extracts. New schedules can be created by Tableau Server Administrators on the Schedules page. For more information, see Create or Modify a Schedule.

**Note:** When a refresh is performed on extracts created in Tableau 10.4 and earlier (that is, a .tde extract), the extract is upgraded to a .hyper extract automatically. While there are many benefits of upgrading to a .hyper extract, your users won’t be able to open the extract with earlier versions of Tableau Desktop. For more information, see Extract Upgrade to .hyper Format.

For information on how to refresh flow outputs, see Schedule a Flow Task.

1. When you’re signed in to Tableau Server, select Explore from the left navigation pane, and then, depending on the type of content you want to refresh, select All Workbooks or All Data Sources from the drop-down menu.

2. Select the check box for the workbook or data source you want to refresh, and then select Actions > Refresh Extracts.

3. In the Refresh Extracts dialog, select Schedule a Refresh, and complete the following steps:

   - Select the schedule you want.

   - If available, specify whether you want a full or incremental refresh.
A full refresh is performed by default. Incremental refresh is available only if you configured for it in Tableau Desktop before publishing the extract. For more information, see Refreshing Extracts in the Tableau Help.

- Click **Schedule Refreshes**.

![Refresh Extracts](image)

Quick Start: Refresh Extracts on a Schedule

For published workbooks that connect to data extracts, you can set up the server to refresh the extracts on a recurring schedule, so all workbooks connected to them always show the most up-to-date data.

To schedule refreshes you need to have administrator or data owner permissions.

**Note:** This topic applies to extracts published to Tableau Server. For Tableau Online, how you refresh extracts depends on the underlying data they connect to. For more information, see Keep Data Fresh.

1 Set up a schedule on the server

Sign in to the server, go to the **Schedules** page, and click **New Schedule**.
Tableau provides a few refresh schedules. You create additional schedules you need.

2 Enable scheduled extract refreshes and failure emails

As a server or site administrator, you can enable schedules, as well as email notification when extract refreshes fail.

Select **Settings**, and then go to the **General** page.

- Under Email Notification, select **Send email to data source and workbook owners when scheduled refreshes fail**.

- Under **Embedded Credentials**, select both check boxes to allow publishers to
embed credentials and schedule extract refreshes.

3 Publish a workbook with an extract

In Tableau Desktop, select Server > Publish Workbook. Sign in to the server if you’re not already. In the Publish Workbook to Tableau Server dialog box, click Schedules & Authentication. Under Extract Schedule, select the schedule from the list.
If the original data requires authentication, you will also need to select how you want people to access it.

4 Monitor refresh performance

You can monitor scheduled tasks by viewing **Background Tasks for Extracts** on the **Status** page.
Automate Refresh Tasks

You can associate extract refresh tasks with schedules in Tableau Server to automate refreshing extracts. You can also automate extract refreshes using tabcmd, a command line utility that you can download for use with Tableau Server. In particular, you can use the `refreshextracts` command in combination with other commands in your own script. For example:

```
tabcmd login - http://mytabserver -u jsmith -p P@ssw0rd!
refreshextracts --datasource salesq4
```

**Note:** When an extract refresh is performed on extracts created in Tableau 10.4 and earlier (that is, a .tde extract), the extract is upgraded to .hyper extract automatically. While there are many benefits of upgrading to a .hyper extract, your users won’t be able open the extract with earlier versions of Tableau Desktop. For more information, see Extract Upgrade to .hyper Format.

For information about downloading the tabcmd utility, see tabcmd.

Handle Extract Refresh Alerts

When Tableau Server cannot complete a scheduled refresh, an alert appears to indicate that the refresh has failed. If a scheduled refresh fails five consecutive times, Tableau Server suspends the refresh. When a refresh is suspended, Tableau Server does not try to run it again until someone takes an action that attempts to correct the cause of the failure.

**Note:** The number of consecutive failures for a refresh is set to five by default, but can be changed by a Tableau Server administrator, using the `backgrounder.failure_threshold_for_run_prevention` option. For more information, see `tsm configuration set Options`.

Tableau Server on Linux Administrator Guide

Tableau Software

Version: 2019.4
You will see the Alerts menu only if an extract refresh failed and you are:

- A system or site administrator
- The author of the workbook or data source that couldn’t be refreshed
- The author of a workbook that connects to a data source that couldn’t be refreshed

When you open the Alerts menu you can see more information about the refresh failure(s):

<image>

When a **Data source** is listed as **Embedded** it means that the data source definition (which includes things like the data source credentials or the database name) is embedded, or resides, within the workbook itself, originally created in Tableau Desktop.

When a data source name or workbook name is listed as the **Data source** (for example, **Data source: sales_data**), it means that the data source is a Tableau Server data source. The data source definition resides on Tableau Server.

In the Data pane on Tableau Desktop, you can determine whether the data source is on Tableau Server or is local. If the data source is on the server, a Tableau icon is displayed next to the data source name instead of a database icon:
Resolving Extract Refresh Problems

To resolve refresh issues, you can take any of these actions, based on the cause indicated in the alert:

- **Errors related to access token validation or user credentials**
  
  You can resolve some extract refresh problems by clicking the **Connection Details** in the alert. Select the check box next to the problematic data source, click **Actions > Edit Connection**, and then enter the missing information. Click **Save** when you're done. After you update the connection information, Tableau Server restarts the refresh schedule.

  If you originally embedded the credentials or other data connection information when you published the workbook or data source from Tableau Desktop, you can also republish the workbook or data source. As part of the publishing process, you can choose to set a new refresh schedule. If you don't choose a new schedule, Tableau Server restarts the existing schedule.

- **Errors that indicate the database was unreachable**

  Confirm that the database is online and that you can sign in to access the data. You can use the **Try again** link in the alert to restart the refresh schedule.

- **Errors when using user filters or impersonation**

  See the **Tableau Knowledge Base**.

  If the problem cannot be corrected by editing the data connection, you will need to resolve it in Tableau Desktop and republish the workbook.
Tip: Administrators can edit data connections at any time on the Data Connections page, accessible from each site by clicking the Content tab and Data Connections.

Edit Connections on Tableau Server

Administrators and data source owners can manage the information that describes how a published data source connects to the underlying data. This information includes the database server name or address, the server port, the database user name, and whether or not the password is embedded in the connection.

Note: Ability to edit connections is determined primarily by your site role, rather than by your permissions on the data source. To edit connections, your site role must be Server Administrator, Site Administrator, or Creator. If your role is Creator, you also must be the data source owner.

1. Sign in to the site that has the data sources you want to modify, and on the Content tab, select Explore > Data sources.

2. Select the name of the data source with the connection you want to update.

   Display filters to search for the data source or narrow the scope of the data source list. The values you type into the Server name and Database username fields are treated as regular expressions.

3. In the Connections view, select the Actions (…) menu for the data source, and then select Edit Connection.
To edit multiple data sources, select the data sources you want to edit, then click the Actions menu and Edit Connection.

4. Update the connection information.

For Server name, if you want to use an IP address, make sure the database or its driver supports that type of connection. If it doesn’t, enter the database server name.

See also Authentication types for Google, Salesforce, and WDC data later in this topic.
5. Click **Save**.

6. Refresh the browser page for your changes to take effect.

**Authentication types for Google, Salesforce, and WDC data**

Google BigQuery, Google Analytics, Salesforce.com, and many web data connector (WDC) connections use the OAuth authentication standard, which uses secure access tokens instead of “raw” user name and password credentials. With OAuth, database credentials do not need to be stored in Tableau, and all users connect through this access token, including Tableau Desktop users who want to create or edit workbooks that connect to the data source.

The following sections describe Google and Salesforce connection options. Web data connector options vary, but all involve signing in through the provider’s web-based sign-in form to establish the access token.
Google authentication options

When you edit Google BigQuery or Google Analytics connections, select either of the following options in the Edit Connection dialog box:

- Select **Embed Google BigQuery (or Google Analytics) credentials in the connection** to authenticate through a designated account, and then select an existing account from the list or select **authenticate account now...** to add a new one.

When you add a new account, the Google sign-in page appears. After you provide your database credentials, Google prompts you to confirm Tableau access to the data. When you click **Accept**, Google returns an access token to use for connecting to the data.

**Note:** If you create extracts of your Google data source, select this first option, so that you can schedule refresh tasks.

- Select **Prompt user for Google BigQuery/Analytics credentials** to require users to connect through their own individual access tokens or sign in each time they connect.

Salesforce.com authentication options

When you edit Salesforce.com connections, you can select any of the following options in the Edit Connection dialog box:

- Select **Embed a Salesforce username and password** to use a traditional authentication method.

- Select **Embed Salesforce credentials in the connection** to use an OAuth connection and schedule refresh tasks, and then select an existing account from the list or click **Add a Salesforce Account** to add a new one.
When you add a new account, the Salesforce.com sign-in page appears. After you provide your database credentials, Salesforce.com prompts you to confirm Tableau access to the data. When you allow Tableau access, Salesforce.com creates an access token through which it connects to the data.

- Select **No Salesforce authentication** to require users to sign in to Salesforce.com each time they connect. (This option does not allow scheduled extract refreshes.)

**Monitor progress**

When you save your changes in the **Edit Connection** dialog box, the dialog displays the progress. If you close the dialog box, the modifications continue to run in the background until completed. Tableau Server will make as many changes as possible. Any failures will be skipped, but they will not impede other changes. For example, if you try to change the server name and add a password to several connections, the server names will be
changed, and the passwords on workbooks will be changed. However, because you cannot add a password to a data source, the passwords for the data sources will not be changed.

For information about checking the progress of these tasks, see Background Tasks for Extracts.

Cube Data Sources

Cube (multidimensional) data sources have certain characteristics that make them unique in Tableau.

Cube data sources do not support pass-through connections. This means that when a cube data source is published, you cannot make a connection from Tableau Server using the data source. It also means you cannot create a workbook using the data source in Tableau Server.

Publishing a cube data source to Tableau Server gives you the ability to store the data source on the server. However, to use the data source, you must download the data source to Tableau Desktop and use it locally. To download a published data source you need:

- The **Download/Save As** permission for the data source. For more information, see Permission capabilities.
- Correct drivers installed and ports opened on computer running Tableau Desktop.

Web Data Connectors in Tableau Server

Web data connectors (WDCs) are web pages that provide a data connection that is accessible over HTTP for data sources that don't already have a connector in Tableau. WDCs allow users to connect to almost any data that is accessible over the web and to create extracts for their workbooks. Data sources for a WDC can include internal web services, JSON data, REST APIs, and other sources that are available over HTTP or HTTPS. Users can create their own WDC or use connectors that were created by others.
For information about how to use a WDC in Tableau Desktop, see Web Data Connector in the Tableau Desktop documentation.

For information about how to create a WDC, see the Web Data Connector documentation on Github.

Before you run connectors on Tableau Server

As a security measure, Tableau Server won't run WDCs unless you approve the connector, as explained in this topic.

**Note:** You must be a server administrator to approve WDCs for use on Tableau Server.

WDCs require your approval because they contain executable code and typically make requests to third-party websites. Before a user can use a WDC with Tableau Server, you must add the domain and port used by the connector to a safe list (whitelist) and also include the domains that a connector can send requests to and receive requests from on a secondary safe list (secondary whitelist). Before you do this, we recommend that you vet and test the connector so that you know what the connector does and what sites it connects to. For more information, see Testing and Vetting Web Data Connectors.

When you add a connector to the safe lists (whitelists), you configure Tableau Server to allow connections to a particular URL where the connector is hosted and from a URL which the connector can query. This is the only way to allow Tableau Server to run WDCs. The connectors can then be hosted on a server inside your organization's firewall or on an external domain. Importing WDCs is not supported for Tableau Server.

Manage connectors in a safe list

To add a WDC to the safe list, use the `tsm data-access web-data-connectors add` command. This command and the related commands described below let you perform the following tasks:
Add WDCs to the safe list and secondary safe list.

- Allow or disallow all WDCs, or WDC refreshes.
- Remove one or more connectors from the safe list.
- List all WDCs on the safe list and secondary safe list.

Updating WDC safe lists requires a server restart

After running any commands that make changes to WDCs, you need to apply your pending changes using the `tsm pending-changes apply` command.

The `pending-changes apply` command displays a prompt to let you know this will restart Tableau Server if the server is running. The prompt displays even if the server is stopped, but in that case there is no restart. You can suppress the prompt using the `--ignore-prompt` option, but this does not change the restart behavior. For more information, see `tsm pending-changes apply`.

Add connectors to the safe list and secondary safe list

To add a connector to the safe list and secondary safe list, use the `tsm data-access web-data-connectors add` command, providing the name, safe list URL, and secondary safe list URLs. You can also manage WDCs using the `web-data-connector-settings Entity`. After running this command, you need to apply your pending changes using the `tsm pending-changes apply` command. A prompt warns you that the command will restart Tableau Server. If Tableau Server is currently running it is restarted, and if it is stopped it is left in a stopped state with no restart.

```
```

Notes on formatting:

- Be sure to use straight quotes (" and "), not curly or "smart" quotes, around the name of the WDC.
For many WDCs, the port specified for the --url option is 443 or 80, but you can check the value for your connector by looking at the data source details on Tableau Server. You must specify the port number as part of the URL if the WDC is using SSL (HTTPS). For example, to use the default port for HTTPS, the URL might look like the following: https://example.com:443/WDC/.

If a WDCs send requests to and receive data from multiple domains, use a comma-delimited list of URL for the --secondary option.

To add an entire domain to the secondary safe list, end the domain URL with a wildcard expression. Use (.*) as the wildcard to indicate the entire domain. Be sure to include the parentheses () as part of the expression.

Allow or disallow WDCs or WDC extract refreshes

To allow or disallow WDCs, or WDC extract refreshes, use the following command before running tsm pending-changes apply:

    tsm data-access web-data-connectors allow

Remove one or more WDCs from the safe list

To remove one or more WDCs from the safe list, use the following command before running tsm pending-changes apply:

    tsm data-access web-data-connectors delete

List all WDCs on the safe list

To list all WDCs on the safe list, use the following command:

    tsm data-access web-data-connectors list

Refresh the extract for a connector

When a user creates a workbook that uses a WDC, Tableau Server creates an extract from the data returned by the connector. If the user then publishes the workbook, the publish process sends the workbook and the data extract to the server.
Tableau can refresh an extract that was created by a WDC, the same as it can refresh any extract. If the connector requires credentials to sign in to the web-based data source, you need to ensure that the credentials are embedded with the data source, and that the WDC is on the safe list for the server. Tableau Server cannot refresh the extract if the connector requires credentials and they are not embedded with the data source. This is because the refresh can occur on a schedule or in some other background context, and the server cannot prompt for credentials.

Currently, there is no way to re-authenticate a data source from Tableau Server directly. If the data source has credentials that expire, or was published without embedding the credentials, the workbook and data extract need to be published again with the new embedded credentials.

If the background process that performs the refresh operation fails, it creates an alert and a log entry that indicates this issue. Users will be able to see that the timestamp on the extract does not change.

To disable refresh for all WDCs, use the `tsm data-access web-data-connectors allow -r false` command.

Troubleshooting

If the server experiences problems with adding connectors to the safe list, you can examine the log files. Be sure to check the log files on both the initial server node and on the other nodes that are running the gateway process. For more information about log files, Server Log File Locations.

If the issue is that Tableau Server will not refresh an extract that was created by a WDC, make sure that the `webdataconnector.refresh.enabled` configuration setting has been set to `true`. If it is set to `false`, run the following command to allow extract refreshes for all WDCs on the server:

```
 tsm data-access web-data-connectors allow -r true
```
Testing and Vetting Web Data Connectors

Web Data Connectors (WDCs) contain JavaScript that typically connects to data on another site. Because of this, you should test and vet web data connectors before users use them as data sources for a workbook, and before you use them with Tableau Server.

This topic includes some suggestions for testing and vetting web data connectors.

Examine the source

The code in a web data connector is in JavaScript, so you can open the file (and any external files that the connector uses) and examine the source code.

Many connectors reference external JavaScript libraries, such as the jQuery library or API libraries for third parties. Validate that the URL for external libraries points to a trusted location for the library. For example, if the connector references the jQuery library, make sure that the library is on a site that is considered standard and safe. If it is practical for you to change the source code of the connector, use HTTPS protocol (https://) to reference external libraries (if the source site supports HTTPS) to help verify the site's authenticity.

To the extent possible, make sure you understand what the code is doing. In particular, try to understand how the code is constructing requests to external sites, and what information is being sent in the request.

Note: Experienced JavaScript programmers often compress (minify) their code to reduce the size of the code for download. Dense blocks of code that use cryptic function and variable names are not uncommon. While this can make it more difficult to examine the code, it is not a sign that the code was written to be deliberately difficult to understand.
Test the web data connector in an isolated environment

If possible, test the web data connector in an environment that is isolated from your production environment and from user computers. For example, add a web data connector to a safe list on a test computer or virtual machine that’s running a version of Tableau Server that is not used for production.

Monitor the traffic created by the web data connector

When you test a web data connector, use a tool like Fiddler, Charles HTTP proxy, or Wire-shark to examine the requests and responses that the connector makes. Make sure that you understand what sites the connector makes requests to and what content the connector is requesting. Similarly, examine the responses and their content to be sure that the connector is not reading data or code that is not directly related to the connector’s purpose.

Test the performance and resource usage of the web data connector

When you test a web data connector, use tools to monitor its CPU and memory usage. Remember that the web data connector will run on Tableau Server, which is an environment in which many processes are already running. You want to make sure that when the connector fetches data, the connector does not have an undue impact on server performance.

Check whether the connector writes to disk. If it does, check how much disk space it occupies, and examine the output to make sure you understand what it's writing and why.

Enable Tableau Catalog

Tableau Catalog discovers and indexes all of the content on your Tableau Online site or Tableau Server, including workbooks, data sources, sheets, and flows. Indexing is used to gather information about the content, or metadata, about the schema and lineage of the content. Then from the metadata, Catalog identifies all of the databases, files, and tables used by the content on your Tableau Online site or Tableau Server.

Catalog is available with the Data Management Add-on. For more information, see Use the Data Management Add-on.
In addition to Catalog, metadata about your content can also be accessed from both the Tableau Metadata API and the Tableau Server REST API using Metadata Methods.

Catalog on Tableau Online

Catalog is automatically enabled when Tableau Online is licensed with the Data Management Add-on.

After your Tableau Online site has been licensed with the Data Management Add-on, the content that already exists on your Tableau Online site is immediately indexed. The time it takes to index the content depends on the amount of content you have. After the content is initially indexed, Catalog monitors newly published content and other changes to assets and continues to index in the background.

Catalog on Tableau Server

As a Tableau Server admin, there are a few things that you need to consider before and while enabling Catalog to ensure optimal performance of Catalog in your Tableau Server environment.

Before enabling Catalog

When Catalog is turned on for the first time, the content that already exists on your Tableau Server is immediately indexed. Catalog utilizes the non-interactive microservices container to index all the content on Tableau Server. The indexing process is compromised of two primary parts: initial ingestion and event monitoring.

The time it takes Catalog to index the content depends on a couple of factors:

- **Amount of content on Tableau Server**: The amount of content is measured by the total number of workbooks, published data sources, and flows published to Tableau Server.
- **Resources available to the non-interactive container**: Specifically, the number of threads, processes, and memory available to support the non-interactive microservices container. For information about the non-interactive microservices container, see Tableau Server Microservice Container topic.
After understanding the factors that impact Catalog ingestion, there may be some adjustments you need to make before enabling Catalog. The recommended process and adjustments to support enabling Catalog are described below.

Step 1: Determine the amount of content on Tableau Server

To determine the amount of content on Tableau Server, do the following:

1. Sign in to Tableau Server using your admin credentials.
2. Go to the Explore page.
3. Click the Top-Level Project drop-down menu and add the numbers next to All Workbooks, All Data Sources, and All Flows together. This is the total amount of content on Tableau Server.

Step 2: Estimate how long ingestion will take

To estimate the time it will take Catalog to ingest content on Tableau Server for the first time, compare your Tableau Server setup to a baseline Tableau Server setup.

For a Tableau Server with the following setup, initial ingestion could take about 6 hours to complete.

<table>
<thead>
<tr>
<th>Components</th>
<th>Baseline values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Content</td>
<td>12,000 workbooks, published data sources, and flows</td>
</tr>
<tr>
<td>Threads</td>
<td>2</td>
</tr>
<tr>
<td>JVM heap size (memory)</td>
<td>64 MB (minimum) - 4 GB (maximum)</td>
</tr>
<tr>
<td>Ingestion</td>
<td>~6 hours</td>
</tr>
</tbody>
</table>

If you have roughly half the content in your Tableau Server environment, initial ingestion might take half the time to complete.

For example: 6,000 (workbooks, published data sources, and flows) + 2 (threads) = ~3 hours (ingestion)
If you have roughly double the content in your Tableau Server environment, initial ingestion might take double the time to complete.

For example: 24,000 (workbooks, published data sources, and flows) + 2 (threads) = ~12 hours (ingestion)

Step 3 (optional): Decrease the time of ingestion and increase memory

If you want to increase the speed of initial ingestion and therefore decrease the time of initial ingestion, you can increase the number of threads allocated to the non-interactive microservices container. When you increase the number of threads, memory must be increased as well.

Threads

As a general rule, the time it takes for Catalog to perform initial ingestion is correlated to the number of allocated threads. Therefore, if you want to reduce the estimated time of ingestion by half, double the allocated threads.

To increase threads, use the `tsm configuration set -k graph-letingestor.-providerEventIngestorClient.connectionPool.maxConnectionPerInstance` command. For more information, see the `tsm configuration` and `tsm configuration set Options` topics.

Memory

When increasing the number of threads allocated to ingestion, you should also increase the JVM heap size (memory) to support the services associated with Catalog. Tableau recommends adding no more than 2 GB of memory per additional thread count added.

To increase memory, use the `tsm configuration set -k non-interactive.vmopts` command. For more information, see the `tsm configuration` and `tsm configuration set Options` topics.

Important: Before increasing threads and memory, consider the following:
The recommendations above are for total number of threads for all nodes, not per node or per instance of the non-interactive container.

Tableau recommends that you progressively increase thread count by only 2 threads at a time to avoid issues with CPU utilization of the Tableau Sever repository (PostgreSQL database) while closely monitoring your Tableau Server environment.

Be aware that when too many threads are allocated to initial ingestion, CPU utilization of the PostgreSQL database might spike and failover. Symptoms to watch for include SQLException errors in the vizportal logs. For more information, see Repository Failover topic.

Step 4: Activate the Data Management Add-on

If not already done, after the necessary adjustments have been made to support optimal Catalog ingestion, you can activate the Data Management Add-on. For more information, see License the Data Management Add-on.

Step 5 (optional): Turn off Catalog capabilities

As part of the Data Management Add-on activation, Catalog capabilities are turned on by default. Because of the indexing process and the estimated time it takes to complete, you might consider temporarily turning off Catalog so that Tableau Server users can’t access Catalog capabilities until they are ready and able to provide complete and accurate results.

To turn off Catalog, follow the procedure below.

1. Sign in to Tableau Server using your Tableau Server admin credentials.
2. From the left navigation pane, click Settings.
3. On the General tab, under Tableau Catalog, clear the Turn on Tableau Catalog check box.

Enabling Catalog

To enable Catalog on Tableau Server, follow the procedures described below.

Step 1: Run the tsm maintenance metadata-services enable command

Run the tsm maintenance metadata-services enable command to enable the Tableau Metadata API. After the Metadata API is enabled, Catalog is automatically turned
on when Tableau Server is licensed with the Data Management Add-on. For more information about running the tsm command, see tsm maintenance.

1. Open a command prompt as an admin on the initial node (where TSM is installed) in the cluster.
2. Run the command: **tsm maintenance metadata-services enable**

**Notes:** When running this command, keep the following points in mind:

- This command stops and starts some services used by Tableau Server, which causes certain functionality, such as the Recommendations capability, to be temporarily unavailable.
- A new index of metadata is created at this time. Running this command any subsequent times will create and replace the previous index.

**Step 2: Monitor ingestion progress**

To ensure that the initial ingestion process is going smoothly, Tableau recommends that you monitor its progress through Tableau Server using the set of procedures below. Using the procedures below you can get the status of initial ingestion by content type.

**Step 1: Get the port number of the non-interactive microservices container**

1. Open a command prompt as admin on the initial node (where TSM in installed) in the cluster.
2. Run the following command to get the port number for the non-interactive microservices container: **tsm topology list-ports**
3. In the results, find **noninteractive:primary** and make note of the port number in the last column. The port number will be used in **Step 3** section below.

**Step 2: Get authentication cookies from browser**

1. Open a browser like Google Chrome and sign in to Tableau Server using your Tableau Server admin credentials.

2. Using the Developer tools option (or something similar), go to the Cookies section, and make note of the values for following cookies: **XSFF-TOKEN** and **workgroup_**
session_id.

Do not close the browser window.

**Step 3: Retrieve ingestion status**

Using the same browser window in **Step 2** section above, do the following:

- Copy the URI above and paste it into the browser’s address bar
- Replace `<your-server>` with your Tableau Server computer name
- Replace `<port>` with the port number you noted at the end of **Step 1** above

http://<your-server>:<port>/relationship-service-war/control/backfill/status

For example, http://10.100.00.0:8850/relationship-service-war/control/backfill/status.

**Note:** Alternatively, you can retrieve ingestion status using Postman by forming an http GET request using the URI above and the following required keys in the request header:

<table>
<thead>
<tr>
<th>Key</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cookie</td>
<td>XSRF-TOKEN=&lt;cookie-value&gt;; workgroup_session_id=&lt;cookie-value&gt;</td>
</tr>
<tr>
<td>Content-Type</td>
<td>application/json</td>
</tr>
</tbody>
</table>

**Step 4: Review results**

The endpoint from **Step 3** above returns a JSON blob that delineates ingestion status by content and asset type. When reviewing the results, note the following:

- A `backfillComplete` status that shows `true` indicates initial ingestion is complete
- A `backfillComplete` status that shows `false` indicates initial ingestion is still in progress

For example:

```
[
{"type":"PublishedDatasource","currentId":
```
Step 3: Configure SMTP Setup

If not already set up for Tableau Server, configure SMTP Setup. SMTP supports sending email to owners who need to be contacted about changes to data. For more information about configuring SMTP, see Configure SMTP Setup.

Step 4 (optional): Turn on Catalog capabilities

If you turned off Catalog capabilities before enabling Catalog in one of the procedures above, you must turn on Catalog to make its capabilities accessible to your users.

To turn on Catalog, follow the procedure below.

1. Sign in to Tableau Server using your Tableau Server admin credentials.
2. From the left navigation pane, click Settings.
3. On the General tab, under Tableau Catalog, clear the Turn on Tableau Catalog check box.

Troubleshoot Catalog or the Metadata API

Timeout limit and node limit exceeded messages

To ensure that Catalog tasks or Metadata API queries that have to return a large number of results don’t take up all Tableau Sever system resources, Catalog implements both timeout and node limits.

**Timeout limit**

When tasks in Catalog or queries in the Metadata API reach the timeout limit, you and your users see the following message:

"Showing partial results, Request time limit exceeded. Try again later." or TIME_LIMIT_EXCEEDED

To resolve this issue, as a Tableau Server admin, you can increase the timeout limit using the tsm configuration set -k metadata.query.limits.time command. For more information, see the tsm configuration and tsm configuration set Options topics.

**Important:** Increasing the timeout limit can utilize more CPU for longer, which can the performance of other processes on Tableau Server.

**Node limit**

When tasks in Catalog or queries in the Metadata API reach the node limit, you and your users see the following message:

NODE_LIMIT_EXCEEDED

To resolve this issue, as a Tableau Server admin, you can increase the node limit using the tsm configuration set -k metadata.query.limits.count command. For more information, see the tsm configuration and tsm configuration set Options topics.

**Important:** Increasing the timeout limit can affect system memory.
Indexing performance

If Catalog is taking longer than expected to index the content on Tableau Server, consider increasing the number of threads allocated to the non-interactive microservices container. You can temporarily disable the Metadata API by running the `tsm maintenance metadata-services disable` command and then following the process described in Enable Tableau Catalog section above.

Catalog is automatically enabled when Tableau Online is licensed with the Data Management Add-on.

After your Tableau Online site has been licensed with the Data Management Add-on, the content that already exists on your Tableau Online site is immediately indexed. The time it takes to index the content depends on the amount of content you have. After the content is initially indexed, Catalog monitors newly published content and other changes to assets and continues to index in the background.

**Disable Tableau Catalog**

As a Tableau Server admin or Tableau Online site admin, you can turn off Catalog capabilities at any time. When Catalog is turned off, the features of Catalog, such as adding data quality warnings or the ability to explicitly manage permissions to database and table assets, are not accessible through Tableau Online or Tableau Server itself. However, Catalog continues to index published content and the metadata is accessible from the Tableau Metadata API and metadata methods in the Tableau Server REST API.

1. Sign in to Tableau Online as a site admin or Tableau Server as a server admin.

2. From the left navigation pane, click **Settings**.

3. On the General tab, under Tableau Catalog, clear the **Turn on Tableau Catalog** check box.
Stop indexing metadata on Tableau Server

To stop indexing the published content on Tableau Server, as a Tableau Server admin, you can disable the Tableau Metadata API. To disable the Metadata API, run the `tsm maintenance metadata-services disable` command. For more information, see `tsm maintenance`.

**Note:** Indexing cannot be stopped for a Tableau Online site.

Use Lineage for Impact Analysis

Lineage requires the Data Management Add-on. Starting in 2019.3, Tableau Catalog is available in the Data Management Add-on to Tableau Online and Tableau Server. When Tableau Catalog is enabled in your environment, you have access to lineage for your data sources, flows, databases, and tables. For more information about Tableau Catalog, see "About Tableau Catalog" in the Tableau Server or Tableau Online Help.

Knowing where your data comes from is key to trusting the data, and knowing who else uses it means you can analyze the impact of changes to data in your environment. The lineage feature in Tableau Catalog helps you do both these things.

Navigate lineage

How you navigate to the Lineage pane depends on what kind of item you’re working with.

To see the lineage for Tableau content such as data sources or flows, from Explore, navigate to and open the item, and then select the Lineage tab.
To see lineage for external assets, such as databases and tables, from **External Assets**, navigate to and select an item from the list. A page opens with information about that table, including the name, type, description, columns, and details about each column. To the right of the table information is the Lineage pane, which shows the lineage for that table.
Lineage shows dependencies in relationship to the lineage anchor, which is the item selected. A lineage anchor can be a database, table, workbook, published data source, or a flow. (In the first example, the anchor is the 2016 Challenge data source and in the second example, it’s the Batters table). All the items below the anchor depend, either directly or indirectly, on the anchor—the outputs or the downstream items. The items above the anchor are the items the anchor is either directly or indirectly dependent on—the inputs or the upstream items.

When you select a column in the table, the lineage is filtered to show only downstream items that depend on the column or upstream inputs to the column.

You can select an upstream or downstream item in the Lineage pane to see its details. For example, when you select Workbooks, the list of workbooks depending on this table appears to the left of the Lineage pane.

From the Lineage pane, you can navigate to any item related to your initial choice, in this case the table, by following the links that interest you.
Lineage and custom SQL connections

When you view the lineage of a connection that uses custom SQL, Catalog doesn’t support showing column information. For more information, see Tableau Catalog support for custom SQL in the Tableau Desktop and Web Authoring Help.

Mismatch between lineage count and tab count

You might notice a mismatch in the count of items between the Tableau Catalog Lineage tool and the tabs in Tableau Server or Tableau Online.

The count mismatch is explained by the fact that each—lineage count vs. tab count—counts items a different way. For example, at any given point in time, Catalog can count only items that are indexed, whereas Tableau Server or Tableau Online counts any items that are published. Other reasons for count differences include whether:

- You have “View” permissions for the item.
- An item is hidden.
- Any fields are used in a workbook.
- An item is directly or indirectly connected to.

Workbook count mismatch example

As an example, here’s how the tab count vs. the lineage count is determined for workbooks.
Connected Workbooks tab counts workbooks that meet both these criteria:

- Connects to the data source (whether or not any fields are actually used in the workbook).
- The user has permissions to view (whether it’s a worksheet, dashboard, or story).

Tableau Catalog Lineage counts workbooks that meet all these criteria:

- Has been indexed by Tableau Catalog.
- Connects to the data source and uses at least one field in the data source.
- Contains worksheets, including dashboards or stories that contain a worksheet, that use at least one field in the data source.

When metadata is blocked because of limited permissions, Catalog still counts the workbook. But instead of seeing some of the sensitive metadata, you see Permissions required. For more information, see "Access metadata about related assets (lineage)" in Manage Permissions for External Assets.
Use email to contact owners

At the end of the lineage is Owners. The list of owners includes anyone assigned as the owner of a workbook, data source, or flow, and anyone assigned as the contact for a database or table in the lineage.

You can email owners to let them know about changes to the data.

1. Select Owners to see the list of people who are impacted by the data in this lineage.
2. Select the owners you want to send a message to.
3. Click Send Email to open the email message box.
4. Enter the Subject and your message in the text box, and click Send.

Set a Data Quality Warning

Data quality warnings are a feature of Tableau Catalog. Starting in 2019.3, Tableau Catalog is available in the Data Management Add-on to Tableau Online and Tableau Server. When Tableau Catalog is enabled in your environment, you can set data quality warnings. For more information about Tableau Catalog, see "About Tableau Catalog" in the Tableau Server or Tableau Online Help.

You can set a warning message on a data asset so that users of that data asset are aware of particular issues. For example, you might want to let users know that the data hasn't been refreshed in two weeks or that a data source has been deprecated. You can set one data quality warning per data asset, such as a data source, database, flow, or table.

Who can see the data quality warning

When you set a warning, the warning is visible to users of the asset and any assets downstream from it. For example, a warning set on a table is visible to users looking at a dashboard with an upstream dependency on that table.

Who can set a data quality warning

To set a data quality warning, you must
How to set a data quality warning

There are several types of data quality warnings you can set on a data asset:

- Warning
- Deprecated
- Stale Data
- Under maintenance

In addition to showing the type of data quality warning, you can include an optional message with more details about the warning.

To set a data quality warning:

1. Select the More actions menu (….) next to the data asset you want to create a warning for, and select **Quality Warning**.
2. Select the **Enable warning** check box.
3. Select the **Warning type** from the drop-down list.
4. (Optional) Enter a message to display to users.
5. Click **Save**.

Note that you can set a data quality warning using REST API. For more information, see **Add Data Quality Warning** in the Tableau REST API Help.

**Remove a data quality warning**

When a warning no longer applies, you can remove it by navigating to the data asset with the warning.

1. Select the More actions menu (…) next to the data asset and select **Quality Warning**.
2. Clear the **Enable warning** check box to remove the warning.
3. Click **Save**.
Manage Dashboard Extensions in Tableau Server

Dashboard extensions are web applications that run in custom dashboard zones and can interact with the rest of the dashboard using the Tableau Extensions API. Dashboard extensions give users the ability to interact with data from other applications directly in Tableau.

Note: You must be a server administrator to enable dashboard extensions on the server, or to block specific extensions from running. You must be a server administrator to add extensions to the safe list and to control the type of data the extensions can access. The server administrator can also configure whether users on the site will see prompts when they add or view extensions in a dashboard. For information about extension security and recommended deployment options, see Extension Security - Best Practices for Deployment.

For information about using dashboard extensions in Tableau, see Use Dashboard Extensions.

Looking for Tableau Online? See Manage Dashboard Extensions in Tableau Online.

Before you run extensions on Tableau Server

Tableau supports two types of dashboard extensions: Network-enabled extensions, which can be hosted on web servers located inside or outside of your local network and have full access to the web, and Sandboxed extensions, which run in a protected environment without access to any other resource or service on the web.

Sandboxed extensions are hosted by Tableau and employ W3C standards, such as Content Security Policy (CSP), to ensure the extension can’t make network calls outside of the hosting Tableau Server. A Sandboxed extension can query data in the dashboard, but it can’t send that data anywhere outside of the sandbox. Sandboxed Extensions are supported in
Tableau 2019.4 and later. By default, Sandboxed extensions are allowed to run if extensions are enabled for the site.

Network-enabled dashboard extensions are web applications and could be running on any computer set up as a web server. This includes local computers, computers in your domain, and third-party web sites. Because Network-enabled extensions could be hosted on third-party sites and could have access to the data in the dashboard, you want to only allow the extensions you trust. See Test Network-enabled extensions for security.

For security, you can use the settings for dashboard extensions on Tableau Server to control and limit the dashboard extensions that are allowed to run.

- By default, Sandboxed extensions are allowed to run if extensions are enabled for the site.
- By default, no Network-enabled extensions are allowed unless they have been explicitly added to the safe list.
- By default, only extensions that use the HTTPS protocol are allowed, which guarantees an encrypted channel for sending and receiving data (the only exception is for http://localhost).
- If the Network-enabled extension requires full data (access to the underlying data) the extension will not be able to run on Tableau Server unless you explicitly add the extension to the safe list and grant the extension access to full data.

Control dashboard extensions and access to data

Server administrators can control a global setting to allow extensions for all sites on the server. Server administrators can also put extensions, including Sandboxed extensions, on a global block list to prevent them from running (see Block specific extensions). By default, all Sandboxed extensions are enabled on the server, but site administrators can choose to override the default and prohibit Sandboxed extensions for the site.
Change the global setting enabling extensions on the server

1. To change this setting for the server, go to Manage All Sites > Settings > Extensions. If the server just has a single site, the global controls appear on the settings page for the site.

2. Under Dashboard Extensions, select or clear the Let users run extensions on this server checkbox. If this option is not selected, extensions are not allowed to run. This global setting overrides the Let users run extensions on this site settings for each site.

Change the default settings for a site

Server administrators can control whether to enable extensions for the site and whether to allow Sandboxed extensions on the site. That is, if extensions are enabled on the server, the default site settings allow Sandboxed extensions to run on the site, provided the extension is not specifically blocked on the server. The default site settings allow Network-enabled extensions to run that appear on the safe list for the site.

1. To change these settings for the site, go to Settings > Extensions.

2. Under Dashboard Extensions, configure these options:
   - Let users run extensions on this site
   - Let Sandboxed extensions run unless they are specifically blocked by a server administrator

Server administrators can add or remove Network-enabled extensions from the safe list for a site. When you add an extension to the safe list, you can control whether to allow the extension to have access to full data. See Add extensions to the safe list and configure user prompts.

Identifying an extension

As a web application, an extension is associated with a URL. You use this URL to test and verify the extension. You also use the URL to add the extension to the safe list to allow full data access, or to the block list to prohibit any access.
If you have the extension manifest file (.trex), an XML file that defines properties for the extension, you can find the URL from the <source-location> element.

```xml
<source-location>
  <url>https://www.example.com/myExtension.html</url>
</source-location>
```

If you have added the extension to the dashboard, you can find the URL from the extension properties. From the More Options menu, click About.

The About dialog box lists the name of the extension, the author of the extension, the website of the author, along with the URL of the extension.
Add extensions to the safe list and configure user prompts

To ensure that users can use Network-enabled extensions that are trusted, you can add them to the safe list for the site.

On the safe list, you can control whether to grant the extension full data access. You can also control whether users will see a prompt asking them to allow the extension access to data. If the extension does not require full data access, you don’t need to add it to the safe list. However, you might want to add an extension to the safe list so that you can configure whether or not users see the prompts. When you hide the prompt from users, the extension can run immediately.

1. Go to Settings > Extensions.

2. Under Enable Specific Extensions, add the URL of the extension. See Identifying an extension.

3. Choose to Allow or Deny the extension Full Data Access.

   Full data access is access to the underlying data in the view, not just the summary or aggregated data. Full data access also includes information about the data sources, such as the names of the connection, fields, and tables. In most cases, if you are adding an extension to the safe list so that it can run, you will also want to allow the extension to have access to full data, if the extension requires it. Before adding extensions to the safe list, be sure to Test Network-enabled extensions for security.

4. Choose to Show or Hide the User Prompts.

   Users see the prompts by default when they are adding an extension to a dashboard, or when they are interacting with a view that has an extension. The prompt tells users details about the extension and whether the extension has access to full data. The prompt gives users the ability to allow or deny the extension from running. You can hide this prompt from users, allowing the extension to run immediately.
Block specific extensions

The default global policy allows all Sandboxed extensions and those Network-enabled extensions that appear on the safe list for a site. Server administrators can keep specific extensions from running by adding them to the block list for the server. If an extension is on the global block list it overrides any settings for the extension on the safe list for a site.

1. To add an extension to the blocked list for the server, go to Manage All Sites > Settings > Extensions. On single-site installations, the block list is on the site Extensions settings page.

2. Under Block Specific Extensions, add the URL of the extension. See Identifying an extension.

Test Network-enabled extensions for security

Dashboard extensions are web applications that interact with data in Tableau using the Extensions API. Network-enabled dashboard extensions could be hosted on web servers inside or outside of your domain, and can make network calls and have access to resources on the Internet. Because of this and the potential vulnerabilities, such as cross-site scripting, you should test and vet Network-enabled dashboard extensions before users use them in dashboards on Tableau Desktop, and before you allow the extensions on Tableau Server.

Examine the source files

Dashboard extensions are web applications and include various HTML, CSS, and JavaScript files, and an XML manifest file (*.trex) that defines the properties in the extension. In many cases, the code for a dashboard extension is publicly available on GitHub and can be examined there or downloaded. In the manifest file (*.trex), you can find the source location, or URL indicated where the extension is hosted, the name of the author, and the web site of the author or company to contact for support. The <source-location> element specifies in the URL, the <author> element, specifies the name of the organization and the web site to contact for support (website="SUPPORT_URL"). The web site is the Get Support link user see in the About dialog box for the extension.
Many dashboard extensions reference external JavaScript libraries, such as the jQuery library or API libraries for third parties. Validate that the URL for external libraries points to a trusted location for the library. For example, if the connector references the jQuery library, make sure that the library is on a site that is considered standard and safe.

All extensions are required to use HTTPS protocol for hosting their extensions. You should examine the source files for the extension to ensure that any reference to external libraries is also using HTTPS or is hosted on the same web site as the extension. The one exception to the requirement of HTTPS is if the extension is hosted on the same computer as Tableau.

To the extent possible, make sure you understand what the code is doing. In particular, try to understand how the code is constructing requests to external sites, and what information is being sent in the request. In particular, check if any user-supplied data is validated to prevent cross-site scripting.

Understand data access

The Tableau Extensions API provides methods that can access the names of the active tables and fields in the data source, the summary descriptions of the data source connections, and the underlying data in a dashboard. If an extension uses any of these methods in a view, the extension developer must declare that the extension requires full data permission in the manifest file (.trex). The declaration looks like the following.

```xml
<permissions>
  <permission>full data</permission>
</permissions>
```

Tableau uses this declaration to provide a prompt to users at run time that gives them the option of allowing this access or not. If the extension uses any one of these four methods, without declaring full-data permission in the manifest file, the extension will load but the method calls will fail.

For information about how an extension accesses data from the dashboard, and the JavaScript methods used, see Accessing Underlying Data in the Tableau Extensions API.
To get a better understanding about what the extension can find out about the data, you can use the DataSources sample dashboard extension (available from the Tableau Extensions API GitHub repository) to see what data is exposed when the `getDatasourcesAsync()` method is called.

Test the extension in an isolated environment

If possible, test the dashboard extension in an environment that is isolated from your production environment and from user computers. For example, add a dashboard extension to a safe list on a test computer or virtual machine that's running a version of Tableau Server that is not used for production.

Monitor traffic created by the dashboard extension

When you test a Network-enabled dashboard extension, use a tool like Fiddler, Charles HTTP proxy, or Wireshark to examine the requests and responses that the extension makes. Make sure that you understand what content the extension is requesting. Examine the traffic to be sure that the extension is not reading data or code that is not directly related to the purpose of the extension.

Creators: Connect to data on the web

Before you can create a new workbook and build a view on the web to analyze your data on the web, you need to connect to your data. Tableau supports connecting to data sources on the web published through Tableau Desktop, or, connecting to data directly through Tableau Online or Tableau Server.

Starting in 2019.3, Tableau Catalog is available in the Data Management Add-on to Tableau Server and Tableau Online. When Tableau Catalog is enabled in your environment, in addition to navigating and connecting to data from Explore, you can navigate and connect to more kinds of data, like databases and tables, from Tableau Catalog. For more information about Tableau Catalog, see "About Tableau Catalog" in the Tableau Server or Tableau Online Help.
**Note:** Data connections created in Tableau Server or Tableau Online are **live connections** only. If you need to use an extract for web authoring, you can publish your data source through Tableau Desktop. To publish through Tableau Desktop, see Publish Data Sources and Workbooks.

### Open the Connect to Data page

On the web, you use the Connect to Data page to access data to connect to. After you sign in to Tableau Server or Tableau Online, you can open this page two ways:

- **Home > Create Workbook**
- **Explore > Create > Workbook**

![Connect to Data page](image)

When you have Catalog enabled in your environment, you can filter the page by databases and tables, as well as data sources.

On the Connect to Data page, the tabs you see depend on the product you have.
Tableau Server

On Tableau Server, select from the following tabs to connect to data: On this site, Files, and Connectors.

Connect to data On this site

1. Select **On this site** to browse to or search for published data sources.
2. Select the data source under **Name** and click the **Connect** button.

**Note:** When you have the Data Management Add-on with Tableau Catalog enabled, you can use **On this site** to connect to databases and tables, as well as data sources.

Connect to files

Tableau supports uploading Excel, text-based data sources (.xlsx, .csv, .tsv), and spatial file formats that only require one file (.kml, .geojson, .topojson, .json, and Esri shapefiles and Esri File Geodatabases packaged in a .zip) directly in your browser. In the **Files** tab of the **Connect to Data** window, connect to a file by dragging and dropping it into the field or clicking "Upload from computer."

Use connectors

From the **Connectors** tab, you can connect to data housed in a cloud database or on a server in your enterprise. You need to supply connection information for each data connection that you make. For example, for most data connections, you need to supply a server name and your sign-in information.

**Supported Connectors** has information on how to connect Tableau to each of these connector types to set up your data source. If the connector you need doesn’t appear in the Connectors tab, you can connect to data through Tableau Desktop and publish your data source to Tableau Online or Tableau Server for web authoring. Learn more about how to **Publish a Data Source** in Tableau Desktop.
When Tableau successfully connects to your data, the Data Source page opens so that you can prepare the data for analysis and begin building your view. To learn more, see Creators: Prepare Data on the Web.

Tableau Server connectors

| Actian Matrix* | Google Drive | MySQL |
| Amazon Aurora | IBM BigInsights | OneDrive |
| Amazon Redshift | IBM DB2 | Oracle |
| Aster Database* | IBM PDA (Netezza)* | Pivotal Greenplum Database |
| Box | MariaDB | PostgreSQL |
| Denodo* | MarkLogic* | SAP Sybase ASE* |
| Dropbox | MemSQL | SAP Sybase IQ* |
| Exasol | Microsoft Azure SQL Data Warehouse | Snowflake |
| Google | | |
| BigQuery** | Microsoft SQL Server | Vertica |
| Google Cloud SQL | MonetDB* | |

*Not available on Linux servers.

**Google BigQuery requires OAuth when creating data sources from the web. Learn more about how server administrators can Set up OAuth for Google.

Tableau Catalog Supported Connectors

Tableau Catalog supports making a connection with a subset of the data connectors that Tableau Server supports. If a data source, database, file, or table is grayed out, you can't con-
nect from Tableau Server. You can, however, connect from the Tableau Desktop Connect pane, if you have the correct permissions.

**Tableau Online**

On Tableau Online, select from the following tabs to connect to data: On this site, Files, Connectors, and Dashboard Starters.

**Connect to data On this site**

1. Select **On this site** to browse to or search for published data sources.
2. Select the data source under **Name** and click the **Connect** button

**Note:** When you have the Data Management Add-on with Tableau Catalog enabled, you can use **On this site** to connect to databases and tables, as well as data sources.

**Connect to files**

Tableau supports uploading Excel or text-based data sources (.xlsx, .csv, .tsv) directly in your browser. In the **Files** tab of the Connect to Data window, connect to an Excel or text file by dragging and dropping it into the field or clicking "Upload from computer."

**Use connectors**

From the **Connectors** tab, you can connect to data housed in a cloud database or on a server in your enterprise. You need to supply connection information for each data connection that you make. For example, for most data connections, you need to supply a server name and your sign-in information.

**Supported Connectors** has information on how to connect Tableau to your data using connectors. If the connector you need doesn’t appear in the Connectors tab, you can connect to data through Tableau Desktop and publish your data source to Tableau Online or Tableau Server for web authoring. Learn more about how to **Publish a Data Source** in Tableau Desktop.
Note: If you’re unable to connect to your data from Tableau Online, check to see if the database is publicly accessible. Tableau Online can only connect to data that’s accessible from the public internet. If your data is behind a private network, you can connect using Tableau Bridge. To learn more, see Publishers: Use Tableau Bridge to Keep Tableau Online Data Fresh.

Tableau Online Connectors

Amazon Aurora  Google BigQuery*  MySQL
Amazon RedShift  Google Cloud SQL  OneDrive
Box  Google Drive  Oracle
Denodo  MemSQL  PostgreSQL
Dropbox  Microsoft Azure SQL Data Warehouse  Snowflake
Exasol  Microsoft SQL Server  Vertica

*Google BigQuery requires OAuth when creating data sources from the web. Learn more about how server administrators can Set up OAuth for Google.

Tableau Catalog Supported Connectors

Tableau Catalog supports making a connection with a subset of data connectors that Tableau Online supports. If a data source, database, file, or table is grayed out, you can’t connect from Tableau Online. You can, however, connect from the Tableau Desktop Connect pane, if you have the correct permissions.

Use Dashboard Starters

On Tableau Online, you can author and analyze data from Oracle Eloqua, Salesforce, ServiceNow ITSM, and QuickBooks Online using Dashboard Starters. On the Dashboard...
**Starter** tab, from the list of pre-built designs, select an option and click **Use Dashboard**. For more information, see **Dashboard Starters for Cloud-based Data Sources**.

**After you connect**

When Tableau connects to your data, the Data Source page opens so that you can prepare the data for analysis and begin building your view. To learn more, see **Creators: Prepare Data on the Web**.

**Keep data fresh in web authoring**

**Update uploaded files in Tableau Online or Tableau Server**: If you manually upload a file (Excel or text) for web authoring, Tableau can’t refresh the file automatically. To update your data, select “Edit Connection” to upload a new version of the file.

**Update file-based published data sources in Tableau Online**: If you have a published data source in Tableau Online (published through Tableau Desktop) that uses file-based data, you can keep it fresh using Tableau Bridge. For more information, see **Expand Data Freshness Options by Using Tableau Bridge**.

**Run Initial SQL**

**Note**: Tableau Prep Builder version 2019.2.2 and later supports using Initial SQL, but doesn't yet support all of the same options supported by Tableau Desktop. For information about using Initial SQL with Tableau Prep Builder, see **Use Initial SQL to query your connections** in the Tableau Prep Builder online help.

When connecting to some databases, you can specify an initial SQL command that will run when a connection is made to the database, for example, when you open the workbook, refresh an extract, sign in to Tableau Server, or publish to Tableau Server. Initial SQL is not run when your refresh your view. Note that this initial SQL is different than a custom SQL connection. A custom SQL connection defines a relation (or table) to issue queries against. For more information, see **Connect to a Custom SQL Query**.

You can use this command to:
Tableau Server on Linux Administrator Guide

- Set up temporary tables to use during the session.
- Set up a custom data environment.

You have the option to add an initial SQL command in the Server Connection dialog box or on the Data Source page.

**Note:** If your data source supports running an initial SQL statement, an *Initial SQL* link appears in the lower-left corner of the Server Connection dialog box. For information about your data source, see *Supported Connectors*.

**To use initial SQL**

1. In the Server Connection dialog box, click *Initial SQL*. Or, on the Data Source page, select *Data > Initial SQL* or *Data > Query Banding and Initial SQL* depending on the database you connect to.

2. Enter the SQL command into the Initial SQL dialog box. You can use the *Insert* drop-down menu to pass parameters to your data source.

![Initial SQL dialog box with example SQL command](image)
Note: Tableau does not examine the statement for errors. This SQL statement is simply sent to the database when you connect.

Your software license may restrict you from using initial SQL with your connection. If you publish to Tableau Server, the server must be configured to allow Initial SQL statements. By default, the server software is configured to allow these statements to run when the workbook is loaded in a web browser.

Administrators can configure server to ignore initial SQL statements by using the `tsm configuration set` command:

```
tsm configuration set -k vizqlserver.initialsql.disabled -v true
```

If the server doesn't allow initial SQL statements, the workbook opens, but the initial SQL commands are not sent.

For more information about the `tsm configuration set` command, see the Tableau Server Help.

Parameters in an initial SQL statement

You can pass parameters to your data source in an initial SQL statement. There are several reasons why this is useful:

- You can configure impersonation using the `TableauServerUser` or `TableauServerUserFull` parameters.
- If your data source supports it, you can set up row-level security (for example, for Oracle VPD or SAP Sybase ASE) to make sure that users see only the data that they are authorized to see.
- You can provide more details in logging, for example, the Tableau version or the workbook name.
The following parameters are supported in an initial SQL statement:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Example of returned value</th>
</tr>
</thead>
<tbody>
<tr>
<td>TableauServerUser</td>
<td>The user name of the current server user. Use when setting up impersonation on the server. Returns an empty string if the user is not signed in to Tableau Server.</td>
<td>jsmith</td>
</tr>
<tr>
<td>TableauServerUserFull</td>
<td>The user name and domain of the current server user. Use when setting up impersonation on the server. Returns an empty string if the user is not signed in to Tableau Server.</td>
<td>domain.lan\jsmith</td>
</tr>
<tr>
<td>TableauApp</td>
<td>The name of the Tableau application.</td>
<td>Tableau Desktop Professional</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Tableau Server</td>
</tr>
<tr>
<td>TableauVersion</td>
<td>The version of the Tableau application.</td>
<td>9.3</td>
</tr>
<tr>
<td>WorkbookName</td>
<td>The name of the Tableau workbook. Use only in workbooks with an embedded data source.</td>
<td>Financial-Anal-ysis</td>
</tr>
</tbody>
</table>

Examples

The following examples show different ways you can use parameters in an initial SQL statement.

- This example sets the security context on Microsoft SQL Server:

  ```sql
  EXECUTE AS USER = [TableauServerUser] WITH NO REVERT;
  ```

- This example shows how, on a DataStax data source, you can use parameters to add detail to logging or to set up a session variable to keep track of the data:
SET TABLEAUVERSION [TableauVersion];

- This example can be used to help set up row-level security for Oracle VPD:

  begin
  
  DBMS_SESSION.SET_IDENTIFIER([TableauServerUser]);

  end;

  **Note:** Oracle PL/SQL blocks require a trailing semicolon to terminate the block. Consult Oracle documentation for the proper syntax.

Defer execution to the server

You can defer an initial SQL statement so that it is executed only on the server. One reason to defer execution to the server is if you don’t have permission to execute the commands that set up impersonation. Use `<ServerOnly></ServerOnly>` tags to enclose the commands to be executed only on the server.

Example:

```
CREATE TEMP TABLE TempTable(x varchar(25));
INSERT INTO TempTable VALUES (1);
<ServerOnly>INSERT INTO TempTable Values(2);</ServerOnly>
```

Security and impersonation

If you use the **TableauServerUser** or **TableauServerUserFull** parameter in an initial SQL statement, you will create a dedicated connection that can’t be shared with other users. This will also restrict cache sharing, which can enhance security, but may also slow performance.

Troubleshoot 'create table' for MySQL and Oracle connections

For MySQL connections, tables are not listed after using initial SQL to create table

When you connect to MySQL, if you run an initial SQL statement like the following, tables might not show because of the way Tableau constructs the query:
CREATE TABLE TestV1.testtable77(testID int);

To resolve this issue, add IF NOT EXISTS to the SQL statement:

CREATE TABLE IF NOT EXISTS TestV1.TestTable(testID int);

For Oracle connections, using initial SQL to create table causes Tableau to stall

When you connect to Oracle and run an initial SQL statement like the following, Tableau is stalled with a spinning wheel because of the way Tableau constructs the query:

CREATE TABLE TEST_TABLE (TESTid int)

To resolve this issue, use the following SQL statement:

BEGIN
EXECUTE IMMEDIATE 'create table test_table(testID int)';
EXCEPTION
WHEN OTHERS THEN NULL;
END;

Create, Interact with, and Embed Views on the Web

You can create and interact with views on Tableau Server. For more information, see the following topics in the Tableau Help for users.

Using Tableau on the Web
Tour your Tableau Site
Edit Views on the Web
Join your Data
Create a Dashboard
Enable Ask Data for Sites and Data Sources

Ask Data is supported for all user roles with direct access to data sources: Creators, Explorers, and Interactors. To make Ask Data available in Tableau Online and Tableau Server, data source owners and Tableau administrators need to follow these simple steps.

**Note:** Ask Data doesn’t support multidimensional cube data sources.

Enable or disable Ask Data for a site

As a site administrator, go to the **General** site settings, scroll down to the **Availability of Ask Data** section, and choose from these options:

- **Enabled by default** enables Ask Data for all data sources by default, but lets specific ones be disabled by data source owners and administrators.

- **Disabled by default** disables Ask Data for all data sources by default, but lets specific ones be enabled by data source owners and administrators. (If you select this option, **Always disable Ask Data** becomes available, which fully disables Ask Data throughout your site.)
Ensure that users can access a data source

To use Ask Data, web-authoring must be enabled for your entire Tableau site, and users must have permission to connect to the individual data source.

If a data source has row-level permissions, those permissions also apply to Ask Data, which won’t recognize secure values or make related statistical recommendations.

Enable or disable Ask Data for a data source

Ask Data is enabled for most data sources by default, but you can disable it for data sources it won’t be used with. For enabled data sources, you can change how often analysis occurs, optimizing system performance.

1. At the top of the data source page, click the Details icon:

   ![Explore / Default / World Indicators](image)

2. In the Ask Data section, click Edit.

3. Click the **Enable for this data source** check box.

4. If you’re enabling Ask Data, choose a data source analysis option:

   - **Automatic** checks for changes every 24 hours and analyzes the data source if it is live, has had an extract refreshed, or has been republished. Choose this option for a data source frequently used with Ask Data, so it will be ready before users query it.

   - **Triggered by user request** analyzes the data source only if it has changed
since a user last accessed Ask Data. Choose this option if the data source changes frequently but users query it with Ask Data only occasionally.

- Disabled analyzes only field names, not values.

5. Click **Save**.

**Automatically Build Views with Ask Data**

Ask Data introduces an entirely new way to interact with your data, letting you type a question and instantly get a response right in Tableau. Answers come in the form of automatic data visualizations, with no need to manually drag-and-drop fields or understand the nuances of your data’s structure.

Ask Data lets you ask sophisticated questions naturally, with support for key analytical concepts like time series and spatial analysis, and an understanding of conversational phrases like “last year”, “earliest”, and “most popular”.

**Ask Data to quickly build a viz**

1. In Tableau Server or Tableau Online, navigate to a data source, and then click **Ask Data** at left.

2. In the text box reading “**Ask about fields in this data source**”, enter terms from the data source fields at left.
Hover over each field to learn more about the data it contains. Unlike the data fields seen in standard Tableau views, these fields can't be dragged to the canvas.

3. As you type, Ask Data automatically completes your entry using likely combinations of data fields and analytical functions. To automatically create a viz, either press Enter, or choose from suggested entries below the text box.

Just like that, you've created a compelling visualization of your data!

Rephrase your question using a conversational style

After Ask Data initially builds a viz, you can quickly revise it by typing conversational phrases in the text box reading "Adjust your question or clear all to start over".
In the text box, do any of the following:

- Change the viz type with phrasing like “as a bar chart”.
- Add functions with phrasing like “by country” or “in December”.
- Replace fields by typing “replace [existing field name] with [new field name]”
- Remove fields by typing “remove [field name]”
- Clear the viz by typing “reset” or “clear”.

Rephrase your question using the interface

You can also rephrase questions by clicking options and data fields in the user interface.

Change the viz type

If the default viz doesn't fully reveal your data, click the menu at upper right, and choose from these supported viz types:

- Bar
- Gantt Bar
- Heat Map
- Histogram
- Line
- Map
- Pie
- Scatter
- Text
- Treemap
Note: To automatically create certain viz types, Ask Data sometimes adds fields like “Number of Records” to your entries.

Customize fields and their data

Ask Data gives you several ways to fine-tune how field values are displayed.

- To switch the fields used for the vertical and horizontal axes, click the Swap Axes button above the text box.
- To change a field or its aggregation type (for example, from sum to average), click the field name in the text box.
- If a field is used as a filter, click the name to specify values or wildcard parameters.

- To adjust a numeric range, click words like “high”, “between”, or “cheap” in the text box.
To adjust a relative date filter, click words like “last”, “previous”, or “next”, or “following”.

To delete a field, hover over it and click the X.
Apply simple calculations

Ask Data supports simple calculations between two measures, which you can apply using these symbols:

+ sums the measures
- produces the difference between them
* multiplies
/ divides

In workbooks you save from Ask Data, these calculations don’t become calculated fields but instead ad hoc calculations on the Columns, Rows, or Marks shelves.

Compare differences over time

Ask Data lets you compare time periods with phrases like “year over year”, “quarter over quarter”, or similar comparisons by month, week, or day. The results appear as difference or percent difference table calculations in workbooks you save from Ask Data.
In the text box, click a difference calculation to choose other fields, aggregation methods, and time periods.
Add sheets with other vizzes

To quickly create multiple different vizzes from a data source, add sheets in Ask Data, and then save them in a new workbook.

At the bottom of the web page, do any of the following:

- Click the Add Sheet icon to the right of named sheets.

![Sheet icons](Image)

- Right-click a sheet name, and choose either Duplicate or Delete.

  (To rename sheets from Ask Data, you need to save them in a new workbook.)

Share links to Ask Data vizzes

You can quickly share Ask Data vizzes without even saving a sheet. Simply click the link icon above the query box, and copy a link viewable by anyone with access to the data source.
Embed Ask Data in a web page

After you ask a question with Ask Data, an Embed Code icon appears in the toolbar. Click it to copy code you can incorporate into any web page.

When web pages with Ask Data embedded in them load, Ask Data opens the data source without a viz, waiting for users to ask a question.

To see Ask Data embedded in a web page, users must have a Tableau license and permission to connect to the data source. They can interact with Ask Data just as they would in the Tableau Online or Tableau Server interface, but they can’t save resulting vizzes.

Save the vizzes in a new workbook

Save vizzes from Ask Data to combine them with other Tableau views and dashboards in a workbook. In a workbook, Ask Data is unavailable, but you gain access to Tableau’s full range of data analysis and visualization tools.

1. Above the viz, click the save icon and choose Save as if you’re saving the viz for the first time, or Save to overwrite a previously saved version.
2. If you chose Save as, select Embed password for data source if people who lack access will be viewing the viz.
Tip: To quickly navigate to the workbook, click the save icon again, and choose Edit workbook.

Tips for success

Use keywords

For example, instead of “I want to see all the countries that these airports are in”, try “by airport and country”.

Use exact wording for attributes and values when possible

For example, if your data source includes Airport Code, Airport Name, and Airport Region fields, specify those by name.

See a ranked list

Ask Data maps terms like “best” and “worst” to Top 1 and Bottom 1, respectively. If you want to see broader rankings, use “high” and “low” instead. For example, enter “houses with low sale prices”.

Surround unusually long values with quotation marks

If you want to analyze long field values that contain line returns, tabs, more than two words, or more than 50 characters, surround them with quotation marks. To improve performance, Ask Data doesn’t index fields of that length.

Inspect a View using Explain Data

Explain Data gives you a new window into your data. Use it to inspect, uncover, and dig deeper into the marks in a viz as you build, explore, and analyze your data.

When you select a mark while editing a view and run Explain Data, Tableau builds statistical models and proposes explanations for the selected mark, including potentially related data from the data source that isn’t used in the current view.
Tableau Server on Linux Administrator Guide

As you build different views, use Explain Data as a jumping-off point to help you explore your data more deeply and ask better questions.

**Note:** Creators and Explorers with editing permissions can use Explain Data when editing a view in Desktop, or editing a view on the web in Tableau Online or Tableau Server.

Steps to use Explain Data

To use Explain Data in a view, you must be able to edit a view in Tableau Desktop, Tableau Online, or Tableau Server.

1. Build a visualization. Make sure it uses a measure that is aggregated with SUM, AVG, COUNT, COUNTD, or AGG.
In Tableau Online or Tableau Server, you will need to open a view for editing (click **Edit** in the toolbar).

![Example of a viz that is ready for Explain Data](image)

2. Select a mark of interest, and then click the **Explain Data** icon in the tooltip for the mark. In Tableau Desktop, you can optionally right-click the mark and select **Explain Data** in the context menu.
Explain Data icon in tooltip menu

**Note:** You must select a single mark. Multiple mark selections are not supported. If Explain Data cannot analyze the type of mark selected, the Explain Data icon and context menu command will not be available. For more information, see When Explain Data is not available.

Select a mark and then select Explain Data in the tooltip menu
3. Read the explanations. Explanations are generated for each measure in the current view that can be analyzed.

If multiple explanations are available, click each explanation tab to see the related details.

The explanations window for Explain Data with explanations for a single measure

If multiple measures are available, click each measure tab for more explanations.
The explanations window for Explain Data with explanations for two measures

4. Click the **Open** icon in the top right corner of an explanation viz to open the visualization as a new worksheet and explore the data further.
Open a new worksheet to explore an explanation

Parts of the explanations window

This image is an example of the explanations window for Explain Data, with multiple explanations available.
**A - Selected Mark.** Displays the dimension values of the selected mark to indicate what mark is being described and analyzed.

**B - Measure in Use.** Click to select the measure in use for explanations. Explanations are given for one measure at a time. If multiple measures are available, they are displayed as separate tabs here.

**C - Expected Value Summary.** Describes whether or not the value is unexpected given the other marks in the visualization. Hover over the text in this statement to see details about the expected value range.
If an expected value summary says the mark is lower than expected or higher than expected, it means the aggregated mark value is outside the range of values that a statistical model is predicting for the mark. If an expected value summary says the mark is lower or higher than expected, but within the natural range of variation, it means the aggregated mark value is within the range of predicted mark values, but is lower or higher in that range of values. For related information, see How explanations are analyzed, evaluated, and scored.

D - Explanations List. Displays a list of the possible explanations for the value in the selected mark that Tableau was able to identify. Click an explanation in the list to see a description in the explanation pane on the right.

E - Explanation Description. Displays the selected explanation with a combination of text and vizzes. Click the icon in the top right corner of the viz thumbnail image to add open it as a new sheet in the workbook.
How to use Explain Data in the flow of analysis

Use Explain Data as a jumping-off point for further exploration of your data. The explanations that it generates help you to see the different values that make up the selected marks in a view. It can tell you about the characteristics of the data points in the data source, and how the data might be related (correlations) using statistical modeling. These explanations give you another tool for inspecting your data and finding interesting clues about what to explore next.

While Explain Data can help uncover and describe relationships in your data, it can't tell you what is causing the relationships or how to interpret the data. You are the expert on your data. Your domain knowledge and intuition is key in helping you decide what characteristics might be interesting to explore further using different views.

When reading the explanations created by Explain Data, keep the following points in mind:

- **Consider the shape, size, and cardinality of your data.** While Explain Data can be used with smaller data sets, it requires data that is sufficiently wide and contains enough marks (granularity) to be able to create a model.

- **Don't assume causality.** Correlation is not causation. Explanations are based on models of the data, but are not causal explanations.

  A correlation means that a relationship exists between some data variables, say A and B. You can't tell just from seeing that relationship in the data that A is causing B, or B is causing A, or if something more complicated is actually going on. The data patterns are exactly the same in each of those cases and an algorithm can't tell the difference
between each case. Just because two variables seem to change together doesn't necessarily mean that one causes the other to change. A third factor could be causing them both to change, or it may be a coincidence and there might not be any causal relationship at all.

However, sometimes you have outside knowledge that is not in the data that helps you to identify what's going on. A common type of outside knowledge would be a situation where the data was gathered in an experiment. If you know that B was chosen by flipping a coin, any consistent pattern of difference in A (that isn't just random noise) must be caused by B. For a longer, more in-depth description of these concepts, see the article [Causal inference in economics and marketing](https://www.ted.com/talks/hal_varian_causal_inference_in_economics_and_marketing) by Hal Varian.

**Requirements and considerations for Explain Data**

- **Single marks only**—Explain Data must be run on a single mark. Multiple mark analysis is not supported.

- **Aggregated data**—The view must contain one or more measures that are aggregated using SUM, AVG, COUNT, COUNTD, or AGG (calculated field). At least one dimension must also be present in the view.

- **Single data sources only**—The data must be drawn from a single data source. Explain Data does not work with blended data sources.

- Extracts run faster than live data sources. With live data sources, the process of creating explanations can create many queries (roughly one query per each candidate explanation), which can result in explanations taking longer to be generated.

- The underlying data must be sufficiently wide. An ideal data set has at least 10-20 columns in addition to one (or more) aggregated measures to be explained.

- The explanation of a categorical dimension is easier to interpret if its cardinality is not too high (< 20 categories).
When Explain Data is not available

Sometimes Explain Data will not be available for a selected mark, depending on the characteristics of the data source or the view. If Explain Data cannot analyze the type of mark selected, the Explain Data icon and context menu command will not be available.

Explain Data can't be run in views that use:
- Map coordinate filters
- Blended data sources
- Data sources with parameters
- Data sources that don't support COUNTD or COUNT(DISTINCT ...) syntax, such as Access.
- Filters on aggregate measures
- Disaggregated measures

Explain Data can't be run if you select:
- Multiple marks
- Axis
- Legend
- Grand total
- Trend line or reference line

Explain Data can't be run when the measure to be used for an explanation:
- Isn't aggregated using SUM, AVG, COUNT, COUNTD, or AGG
- Is a table calculation
- Is used in measure values

Explain Data can't offer explanations for a dimension when it is:
- A parameter
- Used in Measure Names and Measure Values

How explanations are analyzed, evaluated, and scored

When you run Explain Data on a mark, a statistical analysis is run on the aggregated mark, and then on possibly related data points from the data source that aren't represented in the current view.
Explain Data first predicts the value of a mark using only the data that is present in the visualization. Next, data that is in the data source (but not in the current view) is considered and added to the model. The model determines the range of predicted mark values, which is within one standard deviation of the predicted value.

If an expected value summary says the mark is *lower than expected* or *higher than expected*, it means the aggregated mark value is outside the range of values that a statistical model is predicting for the mark. If an expected value summary says the mark is lower or higher than expected, *but within the natural range of variation*, it means the aggregated mark value is within the range of predicted mark values, but is lower or higher in that range of values.

Possible explanations are evaluated on their explanatory power using statistical modeling. Explanations are listed based on how informative they are; explanations that are more simple with less variability are favored. For each explanation, Tableau compares the expected value with the actual value. Explanations that don’t meet the defined threshold are not listed.

Models used for analysis

Explain Data builds Bayesian models of the data in a view to predict the value of a mark, and then determines whether a mark is higher or lower than expected given the model. Next, it considers additional information, like adding additional columns from the data source to the view, or flagging record-level outliers, as potential explanations. For each potential explanation, Explain Data fits a new model, and evaluates how unexpected the mark is given the new information. Explanations are scored by trading off complexity (how much information is added from the data source) against the amount of variability that needs to be explained. Better explanations are simpler than the variation they explain.

How scoring works

When you run Explain Data for a mark, the explanations window is displayed. If multiple explanations are available, they are displayed in descending order based on a numerical
score given to each potential explanation. Only explanations with the highest scores are displayed. Scoring works differently for different explanation types.

<table>
<thead>
<tr>
<th>Explanation type</th>
<th>Evaluation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Extreme value</td>
<td>Extreme values are aggregated marks that are outliers, based on a model of the visualized marks. The selected mark is considered to contain an extreme value if a record value is in the tails of the distribution of the expected values for the data. The score of an extreme value is determined by looking at the minimum and maximum values that make up the aggregate mark. If the mark becomes less surprising by removing a value, then it receives a higher score. When a mark has an extreme value, it doesn't automatically mean it’s an outlier, or that you should exclude it from the view. That choice is up to you depending on your analysis. The explanation is simply pointing out an interesting extreme value in the mark. For example, it could reveal a mistyped value in a record where a banana cost 10 dollars instead of 10 cents. Or, it could reveal that a particular sales person had a great quarter.</td>
</tr>
</tbody>
</table>
| Number of records / Average value of records | This type of explanation is used for aggregate marks that are sums. It explains whether the mark differs from the distribution overall because:  
  - the number of records that make up the mark is higher or lower than the number of records expected for the mark, or  
  - the average value for this mark is higher or lower than expected.  
Because SUM marks are by definition equal to COUNT(X) * AVG(X), the mark can be broken down into a count of values and multiplied by |
Explanation type | Evaluation
---|---
the average value for the mark. This yields two new distributions: a distribution of counts and a distribution of averages. If the selected mark is an outlier, it will either have a count that is an outlier in the count distribution, an average value that is an outlier in the distribution of averages, or both.

This explanation describes whether the sum is interesting because the count is high or low, or because the average is high or low.

Unvisualized and aggregated dimensions

An *unvisualized dimension* is a dimension that exists in the data source, but isn’t currently being used in the view. This type of explanation is used for sums and averages. Aggregated explanations also work on counts.

The model for unvisualized dimensions is created by splitting out marks according to the categorical values of the explaining column, and then building a model with the value that includes all of the data points in the source visualization. For each row, the model attempts to recover each of the individual components that made each mark. The score indicates whether the model predicts the mark better when components corresponding to the unvisualized dimension are modeled and then added up, versus using a model where the values of the unvisualized dimension are not known.

Aggregate dimension explanations explore how well mark values can be explained without any conditioning. Then, the model conditions on values of each column that is a potential explanation. Conditioning on the distribution of an explanatory column should result in a better prediction. The score is basically how much better the prediction becomes.
Create a Subscription to a View or Workbook

Subscriptions email you an image or PDF snapshot of a view or workbook at regular intervals—without requiring you to sign in to Tableau Server or Tableau Online.

**Note:** In Tableau Server, administrators determine whether subscriptions are enabled for a site.

Set up a subscription for yourself or others

When you open a view in Tableau Server or Tableau Online, if you see a subscription icon (ียม) in the toolbar, you can subscribe to that view or to the entire workbook. You can subscribe other users who have permission to view the content if you own a workbook, if you are a project leader with an appropriate site role, or if you are an administrator.

1. From the Explore section of your site, select **All Workbooks** or **All Views**, or open the project that contains the view you want to subscribe to.
2. Open a view either directly, or after opening the containing workbook.

3. In the toolbar above the view, click **Subscribe**.

4. Add the Tableau users or groups you want to receive the subscription. To receive a subscription, users must have the **View** and **Download Image/PDF** permissions. If they use Tableau Server, their accounts must also have email addresses.

**Note:** When you subscribe a group, each user is added individually at the time the subscription is created. If more users are added to the group later, you must re-subscribe the group for those new users to receive the subscription. Likewise, users later removed from the group will not have their subscriptions removed automatically unless their permissions to the subscribed view are removed.
5. Choose whether subscription emails include the current view or the entire workbook.

6. Choose the format for your snapshot: as a PNG image, a PDF attachment, or both.

7. Pick a schedule:
   - For Tableau Server, choose from subscription schedules established by your administrator.
   - For Tableau Online and Tableau Server with custom schedules enabled, click the drop-down arrow to the right of the current settings.

   5 days a week, at 08:00 ▼

   Then specify a custom schedule that sends subscription emails whenever you wish. (The precise delivery time may vary if server load is high.)

   Repeats
   Hourly ▼
   Every:
   Hour ▼
   From: 09:00 ▼ To: 17:00 ▼
   (UTC-08:00) America/Los_Angeles
   On S M T W Th F S

   To change the time zone, click the Time Zone link it to go to your account settings page.

8. To clarify subscription emails, customize the subject line, and add a message.

9. If the view contains data only when high-priority information exists, select Don't send
if view is empty.

10. If you own the workbook, select Subscribe me.

11. Click Subscribe.

When you receive a subscription email, you can select the image (or the link in the message body for PDF subscriptions) to be taken to the view or workbook in Tableau Online or Tableau Server.

Update subscription settings

Update or unsubscribe from a subscription

You can unsubscribe from an existing subscription, or make changes to a subscription’s format, schedule, subject, or empty view mode.

1. Access your Tableau Server or Tableau Online account settings by doing one of the following:
Click Manage my subscriptions at the bottom of a subscription email.

Sign in to Tableau Server or Tableau Online. At the top of the page, select your user icon, and then select My Content.

2. Click Subscriptions.

3. Select the check box next to the view you want to unsubscribe from, click Actions,
and then click **Unsubscribe**, or select the subscription option you'd like to change.

### Resume or delete suspended subscriptions

If a subscription fails more than five times, you'll receive a notification email that your subscription has been suspended. There are a few ways to resume a suspended subscription if you're a subscription owner or administrator:

- From the My Content area of Tableau web pages, an icon appears in the Last update column to indicate that the subscription is suspended. Select ... > **Resume Subscription** to resume.

- From the Subscriptions tab of the affected workbook, an icon appears in the last update column to indicate that the subscription is suspended. Select ... > **Resume Subscription** to resume.

You'll receive an email notification when the subscription is working again.

### See also

- [*Change Subscription Settings*](#) in the Tableau Desktop and Web Authoring Help.

- [*Project-level administration*](#) in the Tableau Online Help, to learn which site roles allow full Project Leader capabilities.

### Use Dashboard Extensions

Extensions let you add unique features to dashboards or directly integrate them with applications outside Tableau. Adding extensions is easy; you incorporate them into dashboard layouts just like other dashboard objects.

Extensions expand dashboard functionality with the help of web applications created by third-party developers. If you're a developer and want to create your own extensions, see the [*Tableau Extensions API documentation*](#) on GitHub.
**Note:** Tableau administrators can turn off dashboard extensions for Tableau Desktop, Tableau Server, and Tableau Online.

The animation below shows an example extension. With other extensions, you'll see different options.

Click the image to replay the animation.

**Add an extension to a dashboard**

To add an extension, you first need to download the related .trex file, which specifies the extension’s properties, including the URL for the web-based application.

1. In a Tableau workbook, open a dashboard sheet.

2. From the **Objects** section, drag **Extension** to the dashboard.
3. In the “Choose an Extension” dialog box, do either of the following:

   - Click My Extensions, and navigate to a .trex file you previously downloaded.
   - Click Extension Gallery, and download a new extension. Then click My Extensions, and navigate to the downloaded .trex file.

4. If prompted, allow or deny the dashboard extension access to data in the workbook. For more information, see Data security, Network-enabled, and Sandboxed extensions.

   If you allow access, follow any on screen instructions for configuring the extension.

   **Note:** Extension objects will appear blank in prints, PDFs, and images of dashboards (including images in subscription emails).

Configure a dashboard extension

Some dashboard extensions provide configuration options that let you customize features.

1. Select the extension in the dashboard, and from the drop-down menu in the upper-right corner, choose Configure.
2. Follow the on-screen instructions to configure the extension.

Reload a dashboard extension

If a dashboard extension becomes unresponsive, you might need to reload it, which is similar to refreshing a web page in a browser.

1. Select the extension in the dashboard, and from the drop-down menu in the upper-right corner, choose **Reload**.

   The dashboard extension is refreshed and set to its original state.

2. If reloading the extension fails to return it to a useable state, try removing it from the dashboard and adding it again.

Data security, Network-enabled, and Sandboxed extensions

Dashboard extensions are web applications that come in two forms:

- *Network-enabled extensions* run on web servers located outside of your local network.
- *Sandboxed extensions* run in a protected environment without access to any other resource or service on the web.

Before adding a Network-enabled extension or viewing a dashboard with one, be certain that you trust the website that hosts it. By default, dashboard extensions use the HTTPS pro-
tocol, which guarantees an encrypted channel for sending and receiving data, and ensures some privacy and security.

For more information about data security when using dashboard extensions, see Extension Security - Best Practices for Deployment.

Allow or deny data access to a Network-enabled extension

Depending on how an extension is designed, it can access either visible data in a view, or full underlying data, table and field names from data sources, and information about data source connections. When you add an extension, or view a dashboard with one, you’re given an opportunity to allow or deny the extension to run and access this data.

If you’re viewing a dashboard with an extension that requires full data access, and that access has been denied, a message appears in place of the extension. If you trust the extension and want to use it, you can reset permissions and allow the extension to run.

1. Select the extension in the dashboard, and from the drop-down menu in the upper-right corner, choose Reset Permissions.

2. Click either Allow to let the extension run and access data, or Deny to prevent the extension from running.
Ensure that JavaScript is enabled in Tableau Desktop

Dashboard extensions interact with data using the Tableau Extensions API library, a JavaScript library. If you want to use extensions, be sure that JavaScript is enabled in the dashboard security settings:


Ensure that extensions run on Tableau Online or Tableau Server

You can add extensions to workbooks you publish from Tableau Desktop or directly in the web-authoring mode of Tableau Online and Tableau Server. A Tableau administrator must allow extensions to run on a site and add Network-enabled extensions to a safe list. Administrators should only allow extensions that you have tested and trust.

If you want to use a dashboard extension on Tableau Online or Tableau Server, direct your administrator to Manage Dashboard Extensions in Tableau Online or Manage Dashboard Extensions in Tableau Server.

Supported web browsers for Sandboxed extensions

Sandboxed extensions run in all browsers supported Tableau Server and Tableau Online except Internet Explorer 11.

Supported versions of Tableau Server for Sandboxed extensions

You can use Sandboxed extensions in Tableau Server 2019.4 and later.

Get support for dashboard extensions

To get help for an extension, you'll need to contact the developer or company who created it.

1. Select the extension in the dashboard, and from the drop-down menu in the upper-right corner, choose About.
2. Click **Get Support** to go to the support page of the extension developer.

Tableau doesn't provide support for extensions or for other programs that interface with the Extensions API. However, you can submit questions and ask for help in the Tableau developer community.
Manage Server

After installing Tableau Server, you can customize and manage your server. For example, you can manage security, licenses, sites, subscriptions and data-driven alerts, and more.

- Security .......................................................... 713
- Preparing for Configuring OAuth Support ........................................ 847
- Configure Settings for Your Data Provider .................................... 848
- Obtain a Client ID and Enable Google APIs .................................. 849
- Configure Tableau Server for Google OAuth .................................... 851
- Create and edit Google data source .............................................. 851
- Managing access tokens .......................................................... 852
- Create a Connected Salesforce App .............................................. 852
- Configure Tableau Server for Salesforce.com OAuth ....................... 854
- Managing access tokens .......................................................... 855
- Create a Connected Intuit App ..................................................... 856
- Configure Tableau Server for QuickBooks Online ......................... 856
- Managing access tokens .......................................................... 857
- Managing credentials centrally .................................................... 858
- See also ............................................................................. 859
- Conflict error ........................................................................ 859
- Manage Licenses ...................................................................... 965
- Add Users to Tableau Server ..................................................... 1005
Security

As a part of managing Tableau Server, you can configure authentication, data security, and network security.

Authentication

Authentication verifies a user’s identity. Everyone who needs to access Tableau Server—whether to manage the server, or to publish, browse, or administer content—must be represented as a user in the Tableau Server repository. The method of authentication may be performed by Tableau Server (“local authentication”), or authentication may be performed by an external process. In the latter case, you must configure Tableau Server for external authentication technologies such as Kerberos, SSPI, SAML, or OpenID. In all cases, whether authentication takes place locally or is external, each user identity must be represented in the Tableau Server repository. The repository manages authorization meta data for user identities.

Looking for Tableau Server on Windows? See Authentication.

Although all user identities are ultimately represented and stored in the Tableau Server repository, you must manage user accounts for Tableau Server in an identity store. There are two, mutually exclusive, identity store options: LDAP and local. Tableau Server supports arbitrary LDAP directories, but it’s been optimized for Active Directory LDAP implementation. Alternatively, if you are not running an LDAP directory, you can use the Tableau Server local identity store. For more information see Identity Store.

As shown in the following table, the type of identity store you implement, in part, will determine your authentication options.
<table>
<thead>
<tr>
<th>Identity Store</th>
<th>Authentication Mechanism</th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Basic SAM Site SAM K-beros Automatic Login Windows (SSPI) OpenID Connect Trusted Auth Mutual SSL</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Local</td>
<td>X X X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>Active Directory</td>
<td>X X X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>LDAP</td>
<td>X X</td>
<td></td>
<td></td>
<td>X</td>
<td>X</td>
</tr>
</tbody>
</table>

Access and management permissions are implemented through site roles. Site roles define which users are administrators, and which users are content consumers and publishers on the server. For more information about administrators, site roles, groups, Guest User, and user-related administrative tasks, see Users and Site Roles for Users.

**Note:** In the context of authentication, it’s important to understand that users are not authorized to access external data sources through Tableau Server by virtue of having an account on the server. In other words, in the default configuration, Tableau Server does not act as a proxy to external data sources. Such access requires additional configuration of the data source on Tableau Server or authentication at the data source when the user connects from Tableau Desktop.
Add-on authentication compatibility

Some authentication methods can be used together. The following table shows authentication methods that can be combined. Cells marked with an "X" indicate a compatible authentication set. Blank cells indicate incompatible authentication sets.

<table>
<thead>
<tr>
<th>Trusted Authentication</th>
<th>Server-wide SAML</th>
<th>Site SAML</th>
<th>Kerberos</th>
<th>Automatic Login Windows (SSPI)</th>
<th>Mutual SSL</th>
<th>OpenID Connect</th>
</tr>
</thead>
<tbody>
<tr>
<td>Trusted Authentication</td>
<td>N/A</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>X</td>
</tr>
<tr>
<td>Server-wide SAML</td>
<td>X</td>
<td>N/A</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Site SAML</td>
<td>X</td>
<td>X</td>
<td>N/A</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Kerberos</td>
<td>X</td>
<td></td>
<td>N/A</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Automatic Login Windows (SSPI)</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mutual SSL</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>OpenID Connect</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
## Client authentication compatibility

<table>
<thead>
<tr>
<th>Clients</th>
<th>Authentication Mechanism</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Basic</td>
</tr>
<tr>
<td>Tableau Desktop</td>
<td>X</td>
</tr>
<tr>
<td>Tableau Prep</td>
<td>X</td>
</tr>
<tr>
<td>Tableau Mobile</td>
<td>X</td>
</tr>
<tr>
<td>Tabcmd</td>
<td>X</td>
</tr>
<tr>
<td>Web Browsers</td>
<td>X</td>
</tr>
</tbody>
</table>

### Local authentication

If the server is configured to use local authentication, then Tableau Server authenticates users. When users sign-in and enter their credentials, either through Tableau Desktop, tabcmd, API, or web client, Tableau Server verifies the credentials.
To enable this scenario, you must first create an identity for each user. To create an identity, you specify a username and a password. To access or interact with content on the server, users must also be assigned a site role. User identities can be added to Tableau Server in the server UI, using tabcmd Commands, or using the REST API.

You can also create groups in Tableau Server to help manage and assign roles to large sets of related user groups (e.g., “Marketing”).

When you configure Tableau Server for local authentication, you can set password policies and account lockout on failed password attempts. See Local Authentication.

**External authentication solutions**

Tableau Server can be configured to work with a number of external authentication solutions.

**NTLM and SSPI**

If you configure Tableau Server to use Active Directory during installation, then NTLM will be the default user authentication method.

When a user logs onto Tableau Server from Tableau Desktop or a web client, the credentials are passed through to Active Directory, which then verifies them and sends an access token to Tableau Server. Tableau Server will then manage user access to Tableau resources based on the site roles stored in the repository.

If Tableau Server is installed on a Windows computer in Active Directory, they you may optionally enable automatic logon. In this scenario, Tableau Server will use Microsoft SSPI to automatically sign in your users based on their Windows username and password. This creates an experience similar to single sign-on (SSO).

Do not enable SSPI if you plan to configure Tableau Server for SAML, trusted authentication, a load balancer, or for a proxy server. SSPI is not supported in these scenarios. See tsm authentication sspi <commands>.
You can configure Tableau Server to use Kerberos for Active Directory. See Kerberos.

SAML

You can configure Tableau Server to use SAML (security assertion markup language) authentication. With SAML, an external identity provider (IdP) authenticates the user's credentials, and then sends a security assertion to Tableau Server that provides information about the user's identity.

For more information, see SAML.

OpenID Connect

OpenID Connect is a standard authentication protocol that lets users sign in to an identity provider (IdP) such as Google. After they've successfully signed in to their IdP, they are automatically signed in to Tableau Server. To use OpenID Connect (OIDC) on Tableau Server, the server must be configured to use the local identity store. Active Directory or LDAP identity stores are not supported with OIDC. For more information, see OpenID Connect.

Mutual SSL

Using mutual SSL, you can provide users of Tableau Desktop, Tableau Mobile, and other approved Tableau clients a secure, direct-access experience to Tableau Server. With mutual SSL, when a client with a valid SSL certificate connects to Tableau Server, Tableau Server confirms the existence of the client certificate and authenticates the user, based on the user name in the client certificate. If the client does not have a valid SSL certificate, Tableau Server can refuse the connection. For more information, see Configure Mutual SSL Authentication.

Trusted authentication

Trusted authentication (also referred to as "Trusted tickets") lets you set up a trusted relationship between Tableau Server and one or more web servers. When Tableau Server
receives requests from a trusted web server, it assumes that the web server has already handled whatever authentication is necessary. Tableau Server receives the request with a redeemable token or ticket and presents the user with a personalized view which takes into consideration the user’s role and permissions. For more information, see Trusted Authentication.

LDAP

You can also configure Tableau Server to use LDAP for user authentication. Users are authenticated by submitting their credentials to Tableau Server, which will then attempt to bind to the LDAP instance using the user credentials. If the bind works then the credentials are valid and Tableau Server grants the user a session.

“Binding” is the handshake/authentication step that happens when a client tries to access an LDAP server. Tableau Server does this for itself when it makes various non-authentication related queries (such as importing users and groups).

You can configure the type of bind you want Tableau Server to use when verifying user credentials. Tableau Server supports GSSAPI and simple bind. Simple bind passes credentials directly to the LDAP instance. We recommend that you configure SSL to encrypt the bind communication. Authentication in this scenario maybe be provided by the native LDAP solution, or with an external process, like SAML.

For more information about planning for and configuring LDAP, see Identity Store and LDAP Configuration Reference.

Other authentication scenarios

- REST API: Signing In and Out (Authentication)
- Mobile device authentication: Single sign-on for Tableau Mobile
- Certificate trust for TSM clients: Connecting TSM clients
- PAM integration for TSM administration: TSM Authentication
Data access and source authentication

You can configure Tableau Server to support a number of different authentication protocols to various different data sources. Data connection authentication may be independent of Tableau Server authentication.

For example, you may configure user authentication to Tableau Server with local authentication, while configuring Kerberos delegation, OAuth, or SAML authentication to specific data sources. See Data Connection Authentication.

Local Authentication

If the server is configured to use local authentication, then Tableau Server authenticates users. When users sign-in and enter their credentials, either through Tableau Desktop, tabcmd, API, or web client, Tableau Server verifies the credentials. Tableau user names stored in the identity store are associated with rights and permissions for Tableau Server. After authentication is verified, Tableau Server manages user access (authorization) for Tableau resources.

To use local authentication, you must configure Tableau Server with a local identity store during Setup. You cannot use local authentication if your Tableau Server has been configured with an external identity store (LDAP, Active Directory, etc).

Password storage

When local authentication is used, the user’s salted and hashed password is stored in the repository. Passwords are never stored directly, rather, the result of salting and hashing the password is stored. Server uses the bcrypt adaptive hashing function.

Configure password settings

After you install Tableau Server with local authentication, you can use Tableau Server Manager (TSM) to configure a number of password-related settings:
Tableau Server on Linux Administrator Guide

- **Password policies**: these policies define the requirement for password structure, such as length, character types, and other requirements.

- **Password expiration**: enable and specify password expiry.

- **Login rate limit**: Tableau Server throttles the time between sign-in attempts after users enter 3 incorrect passwords. Users will need to wait a few seconds before attempting another sign-in. If users continue to enter incorrect passwords, then they must wait for exponentially longer periods of time in between sign-in attempts. By default, the maximum time between sign-in attempts is 60 minutes.

  Lock out account access after too many failed attempts. You can specify how many failed attempts users are allowed to enter before they are locked out.

- **User password reset**: Enable users to reset passwords. Enabling password reset will configure Tableau Server to display a link on the sign-in page. Users who forget passwords or want to reset a password can use the link to initiate a reset-password work flow. Tableau Server must be configured to send email for this feature to operate. See Configure SMTP Setup.

Use the TSM web interface

1. Open TSM in a browser:

   https://<tsm-computer-name>:8850. For more information, see Sign in to Tableau Services Manager Web UI.

2. Click on **User Identity & Access** on the **Configuration** tab and then click **Authentication Method**.

3. Select Local authentication from the drop-down menu to display the password settings.

4. Configure the password settings and then click **Save Pending Changes**.

5. Click **Pending Changes** at the top of the page:
6. Click **Apply Changes and Restart**.

Use the TSM CLI

For the initial configuration of password policies, we recommend that you use the configuration file template below to create a json file. You can also set any single configuration key listed below with the syntax described in tsm configuration set.

1. Copy the following json template to a file. Fill in the key values with your password policy configuration. See the reference section that follows for more information about key options.

   ```json
   {
   "configKeys": {
   
   "wgserver.localauth.policies.mustcontainletters.enabled": false,
   "wgserver.localauth.policies.mustcontainuppercase.enabled": false,
   "wgserver.localauth.policies.mustcontainnumbers.enabled": false,
   "wgserver.localauth.policies.mustcontainsymbols.enabled": false,
   "wgserver.localauth.policies.minimumpasswordlength.enabled": false,
   "wgserver.localauth.policies.minimumpasswordlength.value": 8,
   
   ```
Tableau Server on Linux Administrator Guide

"wgserver.localauth.policies.maximumpasswordlength.enabled": false,

"wgserver.localauth.policies.maximumpasswordlength.value": 255,

"wgserver.localauth.passwordexpiration.enabled": false,

"wgserver.localauth.passwordexpiration.days": 90,

"wgserver.localauth.ratelimiting.maxbackoff.minutes": 60,

"wgserver.localauth.ratelimiting.maxattempts.enabled": false,

"wgserver.localauth.ratelimiting.maxattempts.value": 5,

"features.PasswordReset": false
}

2. Run the tsm settings import -f file.json to pass the json file with the appropriate values to Tableau Services Manager to configure Tableau Server.

3. Run the tsm pending-changes apply command to apply the changes. See tsm pending-changes apply.

Configuration file reference

This section lists all of the options that can be used to configure password policies.

wgserver.localauth.policies.mustcontainletters.enabled

Default value: false

Require at least one letter in passwords.

wgserver.localauth.policies.mustcontainuppercase.enabled
Default value: false

Require at least one upper-case letter in passwords.

wgserver.localauth.policies.mustcontainnumbers.enabled

Default value: false

Require at least one number letter in passwords.

wgserver.localauth.policies.mustcontainsymbols.enabled

Default value: false

Require at least one special character in passwords.

wgserver.localauth.policies.minimumpasswordlength.enabled

Default value: false

Enforce minimum-length passwords.

wgserver.localauth.policies.minimumpasswordlength.value

Default value: 8

The minimum number of characters passwords must have. Enter a value between 4 and 255, inclusive. You must set wgserver.localauth.policies.minimumpasswordlength.enabled to true to enforce this value.

wgserver.localauth.policies.maximumpasswordlength.enabled

Default value: false

Enforce maximum-length passwords.

wgserver.localauth.policies.maximumpasswordlength.value
Default value: 255

The maximum number of characters passwords may have. Enter a value between 8 and 225, inclusive. You must set `wserv-er.localauth.policies.maximumpasswordlength.enabled` to true to enforce this value.

wgserver.localauth.passwordexpiration.enabled

Default value: false

Enforce password expiry.

wgserver.localauth.passwordexpiration.days

Default value: 90

The number of days before a password expires. Enter a value between 1 and 365, inclusive. You must set `wserv-er.localauth.passwordexpiration.enabled` to true to enforce this value.

wgserver.localauth.ratelimiting.maxbackoff.minutes

Default value: 60

Maximum time between sign-in attempts after a user enters multiple incorrect passwords. Enter a value between 5 and 1440, inclusive.

wgserver.localauth.ratelimiting.maxattempts.enabled

Default value: false

Enforce account lock out after 5 incorrect passwords are entered. To change the number of incorrect passwords that will trigger account lock out, you set `wserv-er.localauth.ratelimiting.maxattempts.value`.

wgserver.localauth.ratelimiting.maxattempts.value
The number of incorrect passwords that a user may enter to trigger account lock out. Enter a value between 5 and 100, inclusive. You must set wgserv-er.localauth.ratelimiting.maxattempts.enabled to true to enforce this value.

features.PasswordReset

Default value: false

Enable users to reset passwords.

SAML

SAML (Security Assertion Markup Language) is an XML standard that allows secure web domains to exchange user authentication and authorization data. You can configure Tableau Server to use an external identity provider (IdP) to authenticate users over SAML 2.0. No user credentials are stored with Tableau Server, and using SAML enables you to add Tableau to your organization’s single sign-on environment. Tableau Server supports all SAML 2.0 features including multi-factor authentication (MFA), forcing password change on first login, password complexity requirements, account lockout, and more.

You can use SAML server wide, or you can configure sites individually. Here’s an overview of those options:

- **Server-wide SAML authentication.** A single SAML IdP application handles authentication for all Tableau Server users. Use this option if your server has only the Default site.

- **Server-wide local authentication and site-specific SAML authentication.** In a multi-site environment, users who are not enabled for SAML authentication at the site level can sign in using local authentication.

- **Server-wide SAML authentication and site-specific SAML authentication.** In a multi-site environment, all users authenticate through a SAML IdP configured at the
site level, and you specify a server-wide default SAML IdP for users that belong to multiple sites.

If you want to use site-specific SAML, you must configure server-wide SAML before you configure individual sites. Server-side SAML does not need to be enabled for site-specific SAML to function, but it must be configured.

User authentication through SAML does not apply to permissions and authorization for Tableau Server content, such as data sources and workbooks. It also does not control access to underlying data that workbooks and data sources connect to.

**Note:** Tableau Server supports both service provider initiated and IdP initiated SAML in browsers only. Connections from Tableau Desktop or the Tableau Mobile app require that the SAML request be service provider initiated.

The following image shows the steps to authenticate a user with single sign-on in a typical service provider initiated flow:
1. User navigates to the Tableau Server sign-in page or clicks a published workbook URL.

2. Tableau Server starts the authentication process by redirecting the client to the configured IdP.

3. The IdP requests the user’s username and password from the user. After the user submits valid credentials, the IdP authenticates the user.

4. The IdP returns the successful authentication in the form of a SAML Response to the client. The client passes the SAML Response to Tableau Server.

5. Tableau Server verifies that the username in the SAML Response matches a licensed user stored in the Tableau Server Repository. If a match is verified, then Tableau Server responds to the client with the requested content.

SAML Requirements

Before you configure SAML on Tableau Server, make sure your environment meets the requirements.

Certificate and identity provider (IdP) requirements

To configure Tableau Server for SAML, you need the following:

- **Certificate file.** A PEM-encoded x509 certificate file with a .crt extension. This file is used by Tableau Server, not the IdP. If you have an SSL certificate, it is possible in some circumstances to use the same certificate with SAML. For more information, see Using SSL certificate and key files for SAML later in this article.

- **Certificate key file.** An RSA or DSA private key file that has the .key extension. RSA keys must be in PKCS#1 or PKCS#8 format. Password protection requirements are as follows:
  - The PKCS#1 RSA key file cannot be password protected.
To use a password-protected key file, you must configure SAML with a RSA PKCS#8 file. Note that a PKCS#8 file with a null password is not supported.

- **IdP account that supports SAML 2.0 or later.** You need an account with an external identity provider. Some examples are PingFederate, SiteMinder, and OpenAM.

- **IdP provider that supports import and export of XML metadata.** Although a manually created metadata file might work, Tableau Technical Support cannot assist with generating the file or troubleshooting it.

### SSL Off-loading

If your organization terminates SSL connections from the IdP at a proxy server before sending the authentication request to Tableau Server, then you may need to make a proxy configuration. In this scenario, SSL is "off-loaded" at the proxy server, which means the https request is terminated at the proxy server and then forwarded to Tableau Server over http.

Tableau Server validates the SAML response message returned from the IdP. Since SSL is off-loaded at the proxy, Tableau Server will validate with the protocol that it receives (http), but the IdP response is formatted with https, so validation will fail unless your proxy server includes the X-Forwarded-Proto header set to https. See Configuring Proxies for Tableau Server.

### Using SSL certificate and key files for SAML

If you are using a PEM-encoded x509 certificate file for SSL, you can use the same file for SAML. For SSL, the certificate file is used to encrypt traffic. For SAML, the certificate is used for authentication.

In addition to the requirements listed in Certificate and identity provider (IdP) requirements above, to use the same certificate for both SSL and SAML, the certificate must also meet the following condition to work for SAML:
• Confirm that the certificate includes only the certificate that applies to Tableau Server and not any other certificates or keys.

To do this, you can create a backup copy of the certificate file, and then open the copy in a text editor to review its contents.

User management requirements

When you enable SAML, user authentication is performed outside of Tableau, by the IdP. However, user management is performed by an identity store: either an external identity store (Active Directory or LDAP) or by Tableau Server in a local identity store. For more information about planning for user management with Tableau Server, see Identity Store.

When you configure the identity store during Setup, you must select the option that reflects how you want to use SAML. If you want to use site-specific SAML, you must configure server-wide SAML before you configure individual sites.

• For site-specific SAML: If you have multiple sites on Tableau Server and want to set up each site for a particular IdP or IdP application (or configure some sites not to use SAML), configure Tableau Server to manage user with a local identity store. For site-specific SAML, Tableau Server relies on the IdP for authentication and does not use passwords.

• For server-wide SAML: If you configure server-wide SAML with a single IdP, you can configure Tableau Server to use the local identity store or an external identity store.

• Server-wide SAML authentication and site-specific SAML authentication. In a multi-site environment, all users authenticate through a SAML IdP configured at the site level. In this scenario, you specify a server-wide default SAML IdP for users who belong to multiple sites. To configure this scenario, Tableau Server must be configured with a local identity store.
SAML compatibility notes and requirements

- **Matching usernames:** The user name stored in Tableau Server must match the user name stored in the IdP. For example, if the user name for Jane Smith is stored in PingFederate as `jsmith`, it must also be stored in Tableau Server as `jsmith`.

  If you are configuring SAML as part of the initial Tableau Server setup, make sure the account you plan to use exists in your IdP before you run setup. During Tableau Server setup you create the server administrator account.

  If you use an Active Directory or LDAP external identity store and you are running in multiple domains (that is, users belong to multiple domains, or your Tableau Server installation includes multiple domains), the IdP must send both the username and domain for a user, and they must match exactly in Tableau Server. Although these can be sent as either `domain\username` or `username@domain.com`, we recommend using `domain\username`.

  For more information, see User Management in Active Directory Deployments.

- **Default signature algorithm:** Tableau Server uses SHA1 signature algorithm. Many IdPs default to SHA256. A mismatch between the IdP and Tableau Server signature algorithms will cause SAML authentication failures. To change to SHA256, set the `SHA256Enabled` configuration entity to `true` during your initial configuration. See `samlSettings` Entity. You can also change to SHA256 by running the following TSM command:
tsm configuration set -k wgserver.saml.sha256 -v true

- **External authentication types**: Tableau Server supports using one external authentication type at a time.

- **Mutual SSL**: Tableau Server does not support mutual SSL (two-way SSL) and SAML together. If you want to use mutual SSL, you can configure it on the IdP.

- **Encryption and site-specific SAML assertions**: Although Tableau Server does not support encrypted SAML assertions from the IdP, all SAML requests and responses are sent over HTTPS.

- **User identity in Tableau Server for tabcmd users**: As described in User management requirements section above, to use tabcmd, you must sign in as a user defined on the server. You cannot use SAML accounts with tabcmd.

- **Using SAML SSO with Tableau clients**: By default, both Tableau Desktop and the Tableau Mobile app allow SAML authentication.

  If your IdP does not support this functionality, you can disable SAML sign-in for Tableau clients using the following commands:

  tsm authentication saml configure --desktop-access disable
  tsm authentication saml configure --mobile-access disable

  **Note**: The --mobile-access disable option is ignored by devices running Tableau Mobile app version 19.225.1731 and higher. To disable SAML for devices running these versions you must disable SAML as a client login option on Tableau Server.

  For more information, see tsm authentication saml <commands>.

- **Distributed installations**: TSM versions of Tableau Server (2018.2 and newer) use
Tableau Server on Linux Administrator Guide

the Client File Service to share files in a multi node cluster. After you have configured SAML on the initial node in your cluster, the Client File Service will distribute certificate and key files to the other nodes.

- **Login URL**: For users to be able to sign in, your IdP must be configured with SAML Login endpoint that sends a POST request to the following URL:


- **Logout URL**: To enable users to sign out after signing in with SAML (single logout, or SLO), your IdP must be configured with a SAML Logout endpoint that sends a POST request to the following URL:


  SLO is only supported for server-wide SAML. SLO is not supported for site-specific SAML.

- **Post-logout redirect URL**: By default, when a user signs out of Tableau Server, the sign-in page is displayed.

  To display a different page after sign-out, use the `tsm authentication saml configure` command with the `-su` or `--signout-url` option.

  - To specify an absolute URL, use a fully-qualified URL starting with `http://` or `https://`, as in this example:

    tsm authentication saml configure -su https://example.com

  - To specify a URL relative to the Tableau Server host, use a page starting with a `/` (slash):
tsm authentication saml configure -su /our-
logoutpage.html

- **Active Directory Federation Service (AD FS):** You must configure AD FS to return additional attributes for Tableau authentication with SAML. The Name ID and username attributes can be mapped to the same AD attribute: **SAM-Account-Name**.

Using SAML SSO with Tableau client applications

Tableau Server users with SAML credentials can sign in to the server from Tableau Desktop or the Tableau Mobile app. For full compatibility, we recommend that the Tableau client application version matches that of the server. To connect using site-specific SAML, users must run version 10.0 or later of the Tableau client application.

Connecting to Tableau Server from Tableau Desktop or Tableau Mobile uses a service provider (SP) initiated connection.

**Redirecting authenticated users back to Tableau clients**

When a user signs in to Tableau Server, Tableau Server sends a SAML request (AuthnRequest) to the IdP, which includes the Tableau application’s **RelayState** value. If the user has signed in to Tableau Server from a Tableau client such as Tableau Desktop or Tableau Mobile, it’s important that the RelayState value is returned within the IdP’s SAML response back to Tableau.

When the RelayState value is not returned properly in this scenario, the user is taken to their Tableau Server home page in the web browser, rather than being redirected back to the application they signed in from.

Work with your Identity Provider and internal IT team to confirm that this value will be included as part of the IdP’s SAML response, and then preserved by any network appliance (such as a proxy or load balancer) that resides between your IdP and Tableau Server.
As part of SAML configuration, you exchange XML metadata between Tableau Server and the IdP. This XML metadata is used to verify a user's authentication information when the user initiates the Tableau Server sign-in process.

Tableau Server and the IdP each generates its own metadata. Each set of metadata must contain the information described in the following list. If either set is missing information, errors can occur when you configure SAML or when users try to sign in.

- **HTTP POST**: Tableau Server accepts only HTTP POST requests for SAML communications. HTTP Redirect is not supported.

- With the Binding attribute set to HTTP-POST, the SAML metadata that Tableau Server and the IdP each export must contain the following elements.

  - The element that specifies the URL that the IdP redirects to after successful authentication. This is required in the Service Provider metadata, not the Identity Provider metadata.

    ```xml
    <md:AssertionConsumerService Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST"
    Location="https://<tableau-server- >/wg/saml/SSO/index.html index="0" isDefault="true"/>
    
    For Site SAML, the Location endpoint is /samlser-
    vice/public/sp/meta-data?alias=<site alias>.
    
    - The logout endpoint element appears in Tableau Server metadata and specifies the URL that the IdP will use for Tableau Server's logout endpoint. If this element is not in the IdP metadata, Tableau Server cannot negotiate a logout endpoint with the IdP and the SAML Logout feature will not be available within the Tableau Server:
SLO is only supported for server-wide SAML. SLO is not supported for site-specific SAML.

- Verify that the metadata XML from the IdP includes a **SingleSignOnService** element, in which the binding is set to **HTTP-POST**, as in the following example:

```
<md:SingleSignOnService Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST"
Location="https://SERVER-NAME:9031/idp/SSO.saml2"/>
```

- This element should appear in IdP metadata and specifies the URL that Tableau Server will use for the IdP’s logout endpoint.

```
<md:SingleLogoutService Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST"
Location="https://SERVER-NAME:9031/idp/slo"/>
```

**Attribute named **username**: You must configure the IdP to return an assertion that includes the **username** attribute in the **saml:AttributeStatement** element. The assertion’s attribute type must be **xs:string** (it should not be typed as **xs:any**).

The following example shows what this might look like.

```
<saml:Assertion assertion-element-attributes>
  <saml:Issuer>issuer-information</saml:Issuer>
  <Signature xmlns="http://www.w3.org/2000/09/xmldsig#">
    ...
  </Signature>
</saml:Assertion>
```
Tableau Server on Linux Administrator Guide

By default, Tableau Server will read the username attribute in the AuthNResponse returned from the Idp. However, some IdPs may return a different attribute that is intended to identify the user. In this case, you may need to change the attribute that Tableau Server reads for the username. To authenticate successfully, the attribute’s value that is returned from the IdP must match an actual username value of a Tableau Server user.

To change the SAML attribute that passes the username value, run the following TSM command:
tsm authentication saml map-assertions --user-name <USER-NAME>.

See tsm authentication.

Configure Server-Wide SAML

Configure server-wide SAML when you want all single sign-on (SSO) users on Tableau Server to authenticate through a single SAML identity provider (IdP), or as the first step to configuring site-specific SAML in a multi-site environment.

If you have configured server-wide SAML and are ready to configure a site, see Configure Site-Specific SAML.

The SAML configuration steps we provide make the following assumptions:

- You are familiar with the options for configuring SAML authentication on Tableau Server, as described in the SAML.
- You have verified that your environment meets the SAML Requirements, and obtained the SAML certificate files described in those requirements.

Before you begin

Gather the certificate files and place them on the Tableau Server.

In the Tableau Server folder, create a new folder named SAML, and place copies of the SAML certificate files in that folder. For example:

/var/opt/tableau/tableau_server/data/saml

(Keep the certificate files in a safe location outside of the Tableau Server directory tree as well.)

Note: If you use the same certificate files for SSL, you could alternatively use the existing certificate location for configuring SAML, and add the IdP metadata file to that
directory when you download it later in this procedure. For more information, see Using SSL certificate and key files for SAML in the SAML requirements.

If you are running Tableau Server in a cluster, then the SAML certificates, keys, and metadata file will be automatically distributed across the nodes when you enable SAML.

Use the TSM web interface

This procedure requires that you upload the SAML certificates to TSM so that they are properly stored and distributed in the server configuration. The SAML files must available to the browser on the local computer where you are running the TSM web interface in this procedure.

If you have gathered and saved the SAML files to the Tableau Server as recommended in the previous section, then run the TSM web interface from the Tableau Server computer where you copied the files.

If you are running the TSM web interface from a different computer, then you will need to copy all SAML files locally before proceeding. As you follow the procedure below, browse to the files on the local computer to upload them to TSM.

1. Open TSM in a browser:

   https://<tsm-computer-name>:8850. For more information, see Sign in to Tableau Services Manager Web UI.

2. On the Configuration tab, select User Identity & Access, and then select the Authentication Method tab.
3. For **Authentication Method**, select **SAML**.

4. In the SAML section that appears, complete Step 1, entering the following settings (do not yet select the check box to enable SAML for the server):

   **Tableau Server return URL**—The URL that Tableau Server users will access, such as https://tableau-server.

   Using https://localhost or a URL with a trailing slash (for example, http://tableau_server/) is not supported.

   **SAML entity ID**—The entity ID uniquely identifies your Tableau Server installation to the IdP.

   You can enter your Tableau Server URL again here. If you plan to enable site-specific SAML later, this URL also serves as the base for each site’s unique ID.

   **SAML certificate and key files**—Click **Select File** to upload each of these files.

   After you provide the information required in Step 1, the **Download XML Metadata File** button in Step 2 becomes available. If you are using a passphrase-protected key file, you will need to enter the passphrase with TSM CLI. See the final step in this procedure.

5. Now select the **Enable SAML authentication for the server** check box above
Step 1.

6. Complete the remaining SAML settings.

   a. For Steps 2 and 3, exchange metadata between Tableau Server and the IdP. (Here’s where you might need to check in with the IdP’s documentation.)

      Select Download XML Metadata File, and specify the file location.

      For other IdPs, go to your IdP account to add Tableau Server to its applications (as a service provider), providing the Tableau metadata as appropriate.

      Follow the instructions in the IdP’s website or documentation to download the IdP’s metadata. Save the .xml file to the same location that holds your SAML certificate and key files. For example:

      /var/opt/tableau/tableau_server/data/saml/idp-metadata.xml

   b. Return to the TSM web UI. For Step 4, enter the path to the IdP metadata file, and then click Select File.

   c. For Step 5: In some cases, you may need to change the assertion values in the Tableau Server configuration to match the assertion names that are passed by your IdP.
You can find assertion names in the IdP's SAML configuration. If different assertion names are passed from your IdP, then you must update Tableau Server to use the same assertion value.

**Tip:** “Assertions” are a key SAML component, and the concept of mapping assertions can be tricky at first. It might help to put this in a tabular-data context, in which the assertion (attribute) name is equivalent to a column heading in the table. You enter that “heading” name, rather than an example of a value that might appear in that column.

d. For Step 6, select the Tableau applications in which you want to give users a single sign-on experience.

> **Note:** The option to disable mobile access is ignored by devices running Tableau Mobile app version 19.225.1731 and higher. To disable SAML for devices running these versions you must disable SAML as a client login option on Tableau Server.

e. For the SAML sign-out redirect, if your IdP supports single logout (SLO), enter the page you want to redirect users to after they sign out, relative to the path you entered for the Tableau Server return URL.

f. (Optional) If you are using a PKCS#8 key that is protected with a passphrase, open the TSM CLI and enter the passphrase as follows:

```
tsm configuration set -k wgserver.saml.key.passphrase -v <passphrase>
```

The passphrase will be encrypted and saved. See Manage Server Secrets.

7. Click **Save Pending Changes** after you’ve entered your configuration information.

8. Click **Pending Changes** at the top of the page:
9. Click **Apply Changes and Restart**.

Use the TSM CLI

**Before you begin**

- Go to your IdP’s website or application, and export the IdP’s metadata XML file.

Confirm that the metadata XML from the IdP includes a `SingleSignOnService` element, in which the binding is set to `HTTP-POST`, as in the following example:

```xml
```

**Configure return URL, SAML entity ID, and specify certificate and key files**

1. Open the command prompt shell and configure the SAML settings for the server (replacing placeholder values with your environment path and file names).

```bash
```

For more information, see `tsm authentication saml configure`.

2. If you are using a PKCS#8 key that is protected with a passphrase, enter the
passphrase as follows:

```
$ tsm configuration set -k wgserver.saml.key.passphrase -v <passphrase>
```

3. If SAML is not already enabled on Tableau Server; for example, you’re configuring it for the first time, or you have disabled it, enable it now:

```
$ tsm authentication saml enable
```

4. Apply the changes:

```
$ tsm pending-changes apply
```

The `pending-changes apply` command displays a prompt to let you know this will restart Tableau Server if the server is running. The prompt displays even if the server is stopped, but in that case there is no restart. You can suppress the prompt using the `--ignore-prompt` option, but this does not change the restart behavior. For more information, see `tsm pending-changes apply`.

**Generate Tableau Server metadata and configure the IdP**

1. Run the following command to generate the required XML metadata file for Tableau server.

```
$ tsm authentication saml export-metadata -f <file-name.xml>
```

You can specify a file name, or omit the `-f` parameter to create a default file named `samlmetadata.xml`.

2. On your IdP’s website or in its application:

   - Add Tableau Server as a Service Provider.

   Refer to your IdP’s documentation for information about how to do this. As part of the process of configuring Tableau Server as a Service Provider, you will
import the Tableau Server metadata file you generated from the `export-metadata` command.

- Confirm that your IdP uses `username` as the attribute to verify users.

**Match assertions**

In some cases, you may need to change the assertion values in the Tableau Server configuration to match the assertion names that are passed by your IdP.

You can find assertion names in the IdP’s SAML configuration. If different assertion names are passed from your IdP, then you must update Tableau Server to use the same assertion value.

**Tip:** “Assertions” are a key SAML component, and the concept of mapping assertions can be tricky at first. It might help to put this in a tabular-data context, in which the assertion (attribute) name is equivalent to a column heading in the table. You enter that “heading” name, rather than an example of a value that might appear in that column.

The following table shows the default assertion values and the configuration key that stores the value.

<table>
<thead>
<tr>
<th>Assertion</th>
<th>Default value</th>
<th>Key</th>
</tr>
</thead>
<tbody>
<tr>
<td>Username</td>
<td>username</td>
<td><code>wgserver.saml.idpattribute.username</code></td>
</tr>
<tr>
<td>Display name</td>
<td>displayName</td>
<td><code>wgserver.saml.idpattribute.displayname</code></td>
</tr>
<tr>
<td>Email</td>
<td>email</td>
<td><code>wgserver.saml.idpattribute.email</code></td>
</tr>
<tr>
<td>Domain</td>
<td>(not mapped by default)</td>
<td><code>wgserver.saml.idpattribute.domain</code></td>
</tr>
</tbody>
</table>

To change a given value, run the `tsm configuration set` command with the appropriate key:value pair.
For example, to change the `username` assertion to the value, `name`, run the following command:

```
tsm configuration set -k wgserver.saml.idpattribute.username -v name
```

After you have updated the assertions, you must run the following command to apply the changes:

```
tsm pending-changes apply
```

Optional: Disable client types from using SAML

By default, both Tableau Desktop and the Tableau Mobile app allow SAML authentication.

If your IdP does not support this functionality, you can disable SAML sign-in for Tableau clients using the following commands:

```
tsm authentication saml configure --desktop-access disable
```

```
tsm authentication saml configure --mobile-access disable
```

**Note:** The `--mobile-access disable` option is ignored by devices running Tableau Mobile app version 19.225.1731 and higher. To disable SAML for devices running these versions you must disable SAML as a client login option on Tableau Server.

After you have updated the assertions, you must run the following command to apply the changes:

```
tsm pending-changes apply
```
Test the configuration

1. In your web browser, open a new page or tab, and enter the Tableau Server URL.

   ![Browser window with Tableau Server URL](image)

   The browser redirects you to the IdP’s sign-in form.

2. Enter your single sign-on user name and password.

   ![Sign-in form](image)

   The IdP verifies your credentials and redirects you back to your Tableau Server start page.

Configure SAML with AD FS on Tableau Server

You can configure Active Directory Federation Services (AD FS) as a SAML identity provider, and add Tableau Server to your supported single sign-on applications. When you integrate AD FS with SAML and Tableau Server, your users can sign in to Tableau Server using their standard network credentials.
Prerequisites

Before you can configure Tableau Server and SAML with AD FS, your environment must have the following:

- A server running Microsoft Windows Server 2008 R2 (or later) with AD FS 2.0 (or later) and IIS installed.

- We recommend that you secure your AD FS server (for example, using a reverse proxy). When your AD FS server is accessible from outside your firewall, Tableau Server can redirect users to the sign in page hosted by AD FS.

- SSL certificate encrypted using SHA-2 (256 or 512 bit) encryption, and that meets the additional requirements listed in the following sections:
  - SSL certificate requirements
  - SAML Certificate and identity provider (IdP) requirements

Step 1: Verify SSL connection to AD FS

AD FS requires an SSL connection. If you haven’t done so yet, complete the steps in Configure SSL for External HTTP Traffic to and from Tableau Server, using a certificate that meets the requirements as specified above.

Alternatively, if Tableau Server is configured to work with a reverse proxy or load balancer where SSL is being terminated (commonly referred to as SSL off-loading), then you do not need to configure external SSL.

Step 2: Configure SAML on Tableau Server

Complete the steps in Configure Server-Wide SAML through downloading the Tableau Server metadata to an XML file. At that point, return here and continue to the next section.
Step 3: Configure AD FS to accept sign-in requests from Tableau Server

Note: These steps reflect a third-party application and are subject to change without our knowledge.

Configuring AD FS to accept Tableau Server sign-in requests is a multi-step process, starting with importing the Tableau Server XML metadata file to AD FS.

1. Do one of the following to open the Add Relying Party Trust Wizard:

   **Windows Server 2008 R2:**
   a. Select **Start menu > to Administrative Tools > AD FS 2.0.**
   b. In **AD FS 2.0**, under **Trust Relationships**, right-click the **Relying Party Trusts** folder, and then click **Add Relying Party Trust**.

   **Windows Server 2012 R2:**
   a. Open **Server Manager**, and then on the **Tools** menu, click **AD FS Management**.
   b. In **AD FS Management**, on the **Action** menu, click **Add Relying Party Trust**.

2. In the **Add Relying Party Trust Wizard**, click **Start**.

3. On the **Select Data Source** page, select **Import data about the relying party from a file**, and then click **Browse** to locate your Tableau Server XML metadata file. By default, this file is named **samlspmetadata.xml**.

4. Click **Next**, and on the **Specify Display Name** page, type a name and description for the relying party trust in the **Display name** and **Notes** boxes.

5. Click Next to skip the **Configure Multi-factor Authentication Now** page.

6. Click Next to skip the **Choose Issuance Authorization Rules** page.
7. Click Next to skip the Ready to Add Trust page.

8. On the Finish page, select the Open the Edit Claim Rules dialog for this relying party trust when the wizard closes check box, and then click Close.

Next, you’ll work in the Edit Claim Rules dialog, to add a rule that makes sure the assertions sent by AD FS match the assertions Tableau Server expects. At a minimum, Tableau Server needs an email address. However, including first and last names in addition to email will ensure the user names displayed in Tableau Server are the same as those in your AD account.

1. In the Edit Claim Rules dialog box, click Add Rule.

2. On the Choose Rule Type page, for Claim rule template, select Send LDAP Attributes as Claims, and then click Next.

3. On the Configure Claim Rule page, for Claim rule name, enter a name for the rule that makes sense to you.

4. For Attribute store, select Active Directory, complete the mapping as shown below, and then click Finish.

The mapping is case sensitive and requires exact spelling, so double-check your entries. The table here shows common attributes and claim mappings. Verify attributes with your specific Active Directory configuration.

<table>
<thead>
<tr>
<th>LDAP Attribute</th>
<th>Outgoing Claim Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>SAM-Account-Name</td>
<td>Name ID</td>
</tr>
<tr>
<td>SAM-Account-Name</td>
<td>username</td>
</tr>
<tr>
<td>Given-Name</td>
<td>firstName</td>
</tr>
<tr>
<td>Surname</td>
<td>lastName</td>
</tr>
</tbody>
</table>
If you are running AD FS 2016 or later, then you must add a rule to pass through all claim values. If you are running an older version of AD FS, skip to the next procedure to export AD FS metadata.

1. Click **Add Rule**.
2. Under **Claim rule template**, choose **Pass Through or Filter an Incoming Claim**.
3. Under **Claim rule name**, enter Windows.
4. On the **Edit Rule - Windows** pop-up:
   - Under **Incoming claim type**, select **Windows account name**.
   - Select **Pass through all claim values**.
   - Click **OK**.

Now you will export AD FS metadata that you’ll import to Tableau Server later. You will also make sure the metadata is configured and encoded properly for Tableau Server, and verify other AD FS requirements for your SAML configuration.

1. Export AD FS Federation metadata to an XML file, and then download the file from https://<adfs server name>/FederationMetadata/2007-06/FederationMetadata.xml.

2. Open the metadata file in a text editor like Sublime Text or Notepad++, and verify that it is correctly encoded as UTF-8 without BOM.

   If the file shows some other encoding type, save it from the text editor with the correct encoding.

3. Verify that AD FS uses forms-based authentication. Sign-ins are performed in a browser window, so you need AD FS to default to this type of authentication.

   Edit c:\inetpub\adfs\ls\web.config, search for the tag , and move the line so it appears first in the list. Save the file so that IIS can automatically reload it.

   **Note:** If you don’t see the c:\inetpub\adfs\ls\web.config file, IIS is not installed and configured on your AD FS server.
4. (Optional) This step is required only if AD FS is configured as an IDP for site-specific SAML. This step is not required if AD FS is configured as the IDP for server-wide SAML.

Configure an additional AD FS relying party identifier. This allows your system to work around any AD FS issues with SAML logout.

Do one of the following:

**Windows Server 2008 R2:**

a. In AD FS 2.0, right-click on the relying party you created for Tableau Server earlier, and click **Properties**.

b. On the **Identifiers** tab, in the **Relying party identifier** box, enter `https://<tableauservername>/public/sp/metadata` and then click **Add**.

**Windows Server 2012 R2:**

a. In AD FS Management, in the **Relying Party Trusts** list, right-click on the relying party you created for Tableau Server earlier, and click **Properties**.

b. On the **Identifiers** tab, in the **Relying party identifier** box, enter `https://<tableauservername>/public/sp/metadata` and then click **Add**.

**Note:** AD FS can be used with Tableau Server for a single relying party to the same instance. AD FS cannot be used for multiple relying parties to the same instance, for example, multiple site-SAML sites or server-wide and site SAML configurations.

5. Turn off AD FS assertion encryption for the relying party. Tableau Server does not currently support assertion encryption.
On the AD FS server, use Windows PowerShell to run the following command, replacing `<MyRelyingPartyName>` in the example command below to the name of the AD FS relying party display name:

```
Set-ADFSRelyingPartyTrust -TargetName <MyRelyingPartyName> -EncryptClaims 0
```

**Note:** If you receive the error `Set-ADFSRelyingPartyTrust Cmdlet cannot be found`, you must add the AD FS PowerShell snap-in. At the command prompt type: `Add-PSSnapin Microsoft.Adfs.PowerShell`, and then repeat this step.

**Step 4: Provide AD FS metadata to Tableau Server**

1. Return to the TSM web UI, and navigate to **Configuration > User Identity & Access > Authentication Method** tab.

2. In Step 4 of the SAML configuration window, enter the location of the XML file you exported from AD FS, and select **Upload**.

3. Complete the remaining steps (matching assertions and specifying client type access) as specified in Configure Server-Wide SAML. Save and apply changes.

4. Stop Tableau Server, open TSM CLI, and run the following commands, so that AD FS can properly redirect to Tableau Server after it accepts users’ SAML credentials:

```
tsm configuration set -k wgserver.saml.sha256 -v true
```
3. Apply the changes:

    tsm pending-changes apply

The `pending-changes apply` command displays a prompt to let you know this will restart Tableau Server if the server is running. The prompt displays even if the server is stopped, but in that case there is no restart. You can suppress the prompt using the `--ignore-prompt` option, but this does not change the restart behavior. For more information, see `tsm pending-changes apply`.

Use SAML SSO with Kerberos Database Delegation

In a Windows Active Directory (AD) environment, you can enable SAML single sign-on (SSO) to Tableau Server, along with Kerberos database delegation. This provides authorized users direct access to Tableau Server, as well as to the underlying data defined in their published workbooks and data sources.

Overview of the process
In a typical scenario:

1. One of your Tableau analysts publishes a dashboard to Tableau Server. That dashboard contains a connection to a Hadoop cluster, for example, that is configured to accept Kerberos credentials.

   Then the workbook publisher sends a link to colleagues for review.

2. When a colleague clicks the link, Tableau Server authenticates the user through the SAML SSO process. Then it looks at the user’s authorization scheme, and if allowed, uses the Tableau Server keytab to accesses the underlying database on behalf of the user. This populates the dashboard with the Hadoop data that the user is authorized to see.

**Configure Tableau Server for SAML with Kerberos**

Using SAML with Kerberos works inherently when you complete the processes to enable each separately:

1. Configure Tableau Server for SAML, as described in Configure Server-Wide SAML.

2. Configure Tableau Server and your underlying databases to accept Kerberos credentials, as described in Enable Kerberos Delegation and related articles.

**Configure Site-Specific SAML**

Use site-specific SAML in a multi-site environment when you want to enable single sign-on, and you also use multiple SAML identity providers (IdPs) or IdP applications. When you enable site SAML, you can specify the IdP or IdP application for each site, or configure some sites to use SAML and the others to use the default server-wide authentication method.

If you want all server users to use SAML and sign in through the same IdP application, see Configure Server-Wide SAML.
Prerequisites for enabling site-specific SAML

Before you can enable SAML single sign-on at the site level, complete the following requirements:

- The Tableau Server identity store must be configured for local identity store.
  
  You cannot configure site-specific SAML if Tableau Server is configured with an external identity store such as Active Directory or OpenLDAP.

- Make sure your environment and your IdP meet the general SAML Requirements.
  
  Some features are supported only in server-wide SAML deployments. For example, single logout (SLO) is only supported for server-wide SAML. SLO is not supported for site-specific SAML.

- You must configure server-wide SAML before you configure site-specific SAML. You do not need to enable server-wide SAML, but site-specific SAML requires the server-wide configuration. See Configure Server-Wide SAML.

- Note the location of the SAML certificate files. You will provide this when you configure the server to support site-specific SAML.
  
  For more information, see Put metadata and certificate files in place in the topic on configuring server-wide SAML.

- Add Tableau Server as a service provider to your IdP. You can find this information in the documentation the IdP provides.

- Confirm that the system clocks of the computer hosting the site-SAML IdP and the computer hosting Tableau Server are within 59 seconds of each other. Tableau Server does not have a configuration option to adjust the response skew (time difference) between the Tableau Server computer and the IdP.
Server-wide settings related to site-specific SAML

In the server’s workgroup.yml file, server-wide settings that are used in some way for site-specific SAML include:

- `wgserver.saml.returnurl` and `wgserver.saml.entityid`: In the settings for configuring site-specific SAML, Tableau provides a site-specific return URL and entity ID based on these settings. The site-specific return URL and entity ID cannot be modified.

- `wgserver.saml.domain`, `wgserver.saml.port`, and `wgserver.saml.protocol` are used for SAML requests at the site level.

Server-wide settings `wgserver.saml.maxauthenticationage` and `wgserver.saml.responseskew` do not apply to site-specific SAML.

Configure the server to support site-specific SAML

After you complete the prerequisites listed above, you can run the following commands to configure the server to support site-specific SAML.

1. Configure Server-Wide SAML. At a minimum, you must run the following TSM command (if you have already configured server-wide SAML, skip to Step 2):

   ```
   tsm authentication saml configure --idp-entity-id <tableau-server-entity-id> --idp-return-url <tableau-server-return-url> --cert-file <path-to-saml-certificate.crt> --key-file <path-to-saml-keyfile.key>
   ```

2. Enable site SAML. Run the following commands:

   ```
   tsm authentication sitesaml enable
   tsm pending-changes apply
   ```
About the commands

The `sitesaml enable` command exposes the **Authentication** tab on each site’s **Settings** page in the Tableau Server web UI. After you configure the server to support site SAML, you can continue to Configure SAML for a site to work through the settings on the **Authentication** tab.

The `pending-changes apply` command displays a prompt to let you know this will restart Tableau Server if the server is running. The prompt displays even if the server is stopped, but in that case there is no restart. You can suppress the prompt using the `--ignore-prompt` option, but this does not change the restart behavior. For more information, see `tsm pending-changes apply`.

If you want to review the commands and settings that will be carried out when you run `pending-changes apply`, you can run the following command first:

```
 tsm pending-changes list --config-only
```

Configure SAML for a site

This section takes you through the configuration steps that appear on the **Authentication** page in the Tableau Server web UI. In a self-hosted Tableau Server installation, this page appears only when support for site-specific SAML is enabled at the server level. It is enabled by default in Tableau Online.

**Note:** To complete this process, you will also need the documentation your IdP provides. Look for topics that refer to configuring or defining a service provider for a SAML connection, or adding an application.

**Step 1: Export metadata from Tableau**

To create the SAML connection between Tableau Server and your IdP, you need to
exchange required metadata between the two services. To get metadata from Tableau Server, do either of the following steps. See the IdP’s SAML configuration documentation to confirm the correct option.

- **Select Export metadata** to download an XML file that contains the Tableau Server SAML entity ID, Assertion Consumer Service (ACS) URL, and X.509 certificate.

  The entity ID is site-specific, and based on the server-wide entity ID that you specified when you enabled site SAML on the server. For example, if you specified `https://tableau_server`, you might see the following entity ID for the site:

  https://tableau_server-/samlservice/public/sp/metadat...a-967c-75bdb6e002a0

  You cannot modify the site-specific entity ID or ACS URL that Tableau generates.

- **Select Download signing and encryption certificate** if your IdP expects the required information in a different way. For example, if it wants you to enter the Tableau Server entity ID, ACS URL, and X.509 certificate in separate locations.

  The following image has been edited to show that these settings are the same in Tableau Online and Tableau Server.

<table>
<thead>
<tr>
<th>Export metadata from Tableau Online</th>
<th>Server</th>
</tr>
</thead>
<tbody>
<tr>
<td>Select an option for obtaining metadata required by the Identity Provider (IdP):</td>
<td></td>
</tr>
<tr>
<td>• Export an XML file that contains the metadata</td>
<td>Export metadata</td>
</tr>
<tr>
<td>or</td>
<td></td>
</tr>
<tr>
<td>• Copy the Tableau Online entity ID and ACS URL individually, and download the X.509 certificate and save it as a CER file.</td>
<td></td>
</tr>
<tr>
<td>Tableau Online entity ID</td>
<td></td>
</tr>
<tr>
<td>Asssion Consumer Service URL (ACS)</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>https://tableau_server-/public/sp/identitystore?alias=ec380571-0e8e-480b-85c1-75f8</td>
<td></td>
</tr>
<tr>
<td>https://tableau_server-/public/sp?alias=ec380571-0e8e-480b-85c1-75f8</td>
<td></td>
</tr>
<tr>
<td>Download certificate</td>
<td></td>
</tr>
</tbody>
</table>

**Steps 2 and 3: External steps**
For Step 2, to import the metadata you exported in step 1, sign in to your IdP account, and use the instructions provided by the IdP’s documentation to submit the Tableau Server metadata.

For Step 3, the IdP’s documentation will guide you also in how to provide metadata to a service provider. It will instruct you to download a metadata file, or it will display XML code. If it displays XML code, copy and paste the code into a new text file, and save the file with a .xml extension.

Step 4: Import IdP metadata to the Tableau site

On the Authentication page in Tableau Server, import the metadata file that you downloaded from the IdP or configured manually from XML it provided.

Step 5: Match attributes

Attributes contain authentication, authorization, and other information about a user. In the Identity Provider (IdP) Assertion Name column, provide the attributes that contain the information Tableau Server requires.

- **Username or Email**: (Required) Enter the name of the attribute that stores user names or email addresses.

- **Display name**: (Optional but recommended) Some IdPs use separate attributes for first and last names, and others store the full name in one attribute.

Select the button that corresponds to the way your IdP stores the names. For example, if the IdP combines first and last name in one attribute, select Display
Step 6: Manage users

Select existing Tableau Server users, or add new users you want to approve for single sign-on.

When you add or import users, you also specify their authentication type. On the Users page, you can change users’ authentication type any time after adding them.

For more information, see Add Users to a Site or Import Users and Set the User Authentication Type for SAML.
**Important:** Users that authenticate with site-specific SAML can belong only to one site. If a user needs to access multiple sites, set their authentication type to the server default. Depending on how site-specific SAML was configured by the server administrator, the server default is either local authentication or server-wide SAML.

### Default authentication type for embedded views

Part of enabling SAML on your site is to specify how users access views embedded in web pages.

- **Allow users to choose their authentication type**

  When you select this, two sign-in options appear where a view is embedded: a sign-in button that uses single sign-on authentication and a link to use TableauID as an alternative.

  **Tip:** With this option, users need to know which alternative to choose. As part of notification you send your users after you add them to the single sign-on site, let them know which type of authentication to use for a variety of sign-in scenarios. For example, embedded views, Tableau Desktop, Tableau Bridge, Tableau Mobile, and so on.

- **TableauID**

  This option requires users to sign in using a TableauID even if SAML is enabled on the site. Generally it’s reserved for administrators for troubleshooting issues with embedded views and SAML.

- **Single sign-on with SAML**
If your IdP doesn’t support signing in from an iframe, select **Authenticate in a separate pop-up window**. When a user goes to the web page with the embedded view, the pop-up window appears when they select the sign-in button.

If your IdP supports signing in from an iframe, select **Authenticate using an inline frame (less secure; not supported by all IdPs)**. Iframe embedding can provide a more seamless user experience. For example, if a user is already authenticated with your IdP, and iframe embedding is enabled, the user seamlessly authenticates with Tableau Server when browsing to pages that contain embedded Tableau views.

**Caution:** Because iframes can be vulnerable to **clickjacking** attacks, not all IdPs support signing in through an iframe. With clickjacking, the attacker tries to lure users into clicking or entering content. They do this by displaying the page to attack in a transparent layer over an unrelated page. For Tableau Server, an attacker might try to capture user credentials or to get an authenticated user to change settings. For more information, see [Clickjacking](https://owasp.org/www-project-clickjacking) on the Open Web Application Security Project website.

Troubleshoot SAML

This topic provides information about resolving issues that can occur when you configure SAML authentication.

**SAML and Enable Automatic Logon**

If you are using SAML and if Tableau Server is also configured to use Active Directory, do not also select **Enable automatic logon**. **Enable automatic logon** and SAML cannot both be used on the same server installation.

**HTTP Status 500 error when configuring SAML**

Under some circumstances you might get an HTTP status 500 error and see the following error after enabling SAML and navigating to the Tableau Server URL in a browser:
org.opensaml.saml2.metadata.provider.MetadataProviderException: User specified binding is not supported by the Identity Provider using profile urn:oasis-is:names:tc:SAML:2.0:profiles:SSO:browser

To help resolve this error, make sure of the following:

- The IdP URL for the SSO profile specified in the SAML tab is correct.
- The IdP URL for the SSO profile provided while creating the service provider in the IdP is correct.
- The IdP is configured to use HTTP-POST requests. (Redirect and SOAP are not supported.)

If any of these settings were not correct, make appropriate updates and then perform the SAML configuration steps again, starting with generating and exporting the XML metadata document from Tableau Server.

If these settings are correct, but you still see the error, examine the metadata XML that is produced by Tableau Server and by the IdP, as described in SAML Requirements.

Signing In from the Command Line

SAML is not used for authentication when you sign in to Tableau Server using tabcmd or the Tableau Data Extract command line utility (provided with Tableau Desktop), even if Tableau Server is configured to use SAML. These tools require the authentication configured when Tableau Server was originally installed (either local authentication or AD).

Login fails: failed to find the user

Login fails with the following message:

> Login failure: Identity Provider authentication successful for user <username from IdP>. Failed to find the user in Tableau Server.
This error typically means that there is a mismatch between the usernames stored in Tableau Server and provided by the IdP. To fix this, make sure that they match. For example, if Jane Smith's username is stored in the IdP as jsmith it must be stored in Tableau Server as jsmith.

Login fails: SSL offloading

Logon fails with the following message:

Unable to Sign In - Invalid username or password.

Additionally, the vizportal logs (set to debug mode) contain the following message:

DEBUG com.tableau.core.util.RemoteIP - Found header null in X-FORWARDED-PROTO

**Note:** To log SAML-related events, vizportal.log.level must be set to debug. For more information, see Change Logging Levels.

This combination of messages indicates a misconfiguration of an external proxy server that is offloading SSL for the connection to Tableau Server. To resolve this issue, see the KB article, "Unable to Sign In" and "Invalid username or password" Error With SAML After Upgrading.

SAML Error Log

SAML authentication takes place outside Tableau Server, so troubleshooting authentication issues can be difficult. However, login attempts are logged by Tableau Server. You can create a snapshot of log files and use them to troubleshoot problems. For more information, see Log File Snapshots (Archive Logs).

**Note:** To log SAML-related events, vizportal.log.level must be set to debug. For more information, see Change Logging Levels.
Check for SAML errors in the following files in the unzipped log file snapshot:
\vizportal\vizportal-<n>.log

In Tableau Server 9.0 and later, the application process (vizportal.exe) handles authentication, so SAML responses are logged by that process.

Trailing Slash

On the SAML tab, confirm that the Tableau Server return URL does not end with a trailing slash

Correct: http://tableau_server

Incorrect: http://tableau_server/

Confirm Connectivity

Confirm that the Tableau Server you are configuring has either a routeable IP address or a NAT at the firewall that allows two-way traffic directly to the server.

You can test your connectivity by running telnet on Tableau Server and attempting to connect with the SAML IdP. For example: C:\telnet 12.360.325.10 80

The above test should connect you to the HTTP port (80) on the IdP and you should receive an HTTP header.

Kerberos

Kerberos is a three-way authentication protocol that relies on the use of a trusted third-party network service called the Key Distribution Center (KDC) to verify the identity of computers and provide for secure connections between the computers through the exchange of tickets. These tickets provide mutual authentication between computers or services, verifying that one has permission to access the other.

Tableau Server supports Kerberos authentication in an Active Directory Kerberos environment, with authentication to Tableau Server being handled by Kerberos.
Note: The Kerberos support in Tableau Server is for user authentication. It does not handle internal permissions and authorization related to Tableau Server content, such as workbooks.

How Kerberos works

When you configure Tableau Server for Kerberos in an Active Directory (AD) environment, the AD domain controller also serves as the Kerberos Key Distribution Center (KDC) and issues Ticket Granting Tickets to the other nodes in the domain. Users authenticated by the KDC do not have to authenticate further when connecting to Tableau Server.

The following is a diagram of the authentication workflow.

1. User logs into their Active Directory domain.
2. The Kerberos KDC authenticates the user and sends a Ticket Granting Ticket (TGT) to the user’s computer.
3. The user connects to Tableau Server.
4. Tableau Server authenticates the user.
Server in Tableau Desktop or in a web browser.

Kerberos Requirements

You can configure Kerberos authentication for Tableau Server running in Active Directory environments.

General requirements

- **External Load Balancer/Proxy Server:** If you are going to use Tableau Server with Kerberos in an environment that has external load balancers (ELBs) or proxy server, you need to set these up before you configure Kerberos in the Tableau Server Configuration utility. See Configuring Proxies for Tableau Server.

- **iOS Browser Support:** An iOS user can use Kerberos authentication with mobile Safari if a Configuration Profile specifying the user's Kerberos identity is installed. See Configuring an iOS Device for Kerberos Support in the Tableau Mobile Help. For more information about browser support for Kerberos SSO, see Tableau Client Support for Kerberos SSO.

- **Tableau Server supports constrained delegation for authentication to datasources.** In this scenario, the Tableau data access account is specifically granted rights to the target database SPNs. Unconstrained delegation is not supported.

- **The supported data sources (SQL Server, MSAS, PostgreSQL, Hive/Impala, and Teradata) must be configured for Kerberos authentication.**

- **A keytab file that is configured with the service provider name for the Tableau Server for user authentication.** If you are using Kerberos authentication for datasources then those credentials should be included in the single keytab file that you will specify during Kerberos configuration on Tableau Server.
You must meet the following requirements to run Tableau Server with Kerberos in an Active Directory environment:

- Tableau Server must use Active Directory (AD) for authentication.

- The domain must be an AD 2003 or later domain for Kerberos connections to Tableau Server.

- Smart Card Support: Smart cards are supported when users sign into their workstations with a smartcard and this results in a Kerberos TGT being granted to the user from Active Directory.

- Single-Sign On (SSO): Users must be granted a Kerberos Ticket Granting Ticket (TGT) from Active Directory when they sign into their computers. This is standard behavior for domain-joined Windows computers and standard for Mac computers that use AD as their network account server. For more information on using Mac computers and Active Directory, see Join your Mac to a network account server in the Apple Knowledge Base.

**Kerberos delegation**

For Kerberos delegation scenarios the following are required:

- If the domain is AD 2003 or later, single domain Kerberos delegation is supported. The users, Tableau Server, and backend database must be on the same domain.

- If the domain is AD 2008, there is limited cross domain support. Users from other domains can be delegated if the following conditions are met. Tableau Server and the backend database must be on the same domain, and a two way trust is required between the domain where Tableau Server resides and the user’s domain.

- If the domain is 2012 or later, full cross-domain delegation is supported. AD 2012 R2
Tableau Server on Linux Administrator Guide

is preferred because it has a dialog for configuring constrained delegation, while 2012 non-R2 requires manual configuration.

Understanding Keytab Requirements

Kerberos authentication relies on credentials that are stored in specially formatted files called keytab files. You may need to generate keytab files for your Tableau Server deployment. This topic describes the keytab files that Tableau Server uses to access various services in a typical organization. You may need to generate keytabs for Tableau Server to integrate into the following services:

- User authentication (SSO) in Windows Active Directory
- Datasource delegation
- Operating system
- Directory service

If your organization includes IT professionals who handle identity, authentication, and/or security, then you should work with them to create a plan for generating appropriate keytabs for your Tableau Server deployment.

User authentication (SSO) in Windows Active Directory

If you will be using Active Directory as the identity store for Tableau Server, and you want users to authenticate with Kerberos SSO, then you will need to generate a keytab file for Tableau Server.

<table>
<thead>
<tr>
<th>Tableau is running on...</th>
<th>Need to manually generate a keytab?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Windows in Active Directory domain</td>
<td>Yes</td>
</tr>
<tr>
<td>Linux in Active Directory domain</td>
<td>Yes</td>
</tr>
<tr>
<td>Windows or Linux in non-Active Directory environment</td>
<td>Kerberos SSO is not a supported scenario.</td>
</tr>
</tbody>
</table>

Follow these recommendations (for Windows and Linux versions of Tableau Server):
Create a service account in your directory for Tableau Server.

Create a keytab specifically for the Tableau Server service account. Do not reuse the keytab file that the computer account/OS uses to authenticate. You may use the same keytab for Kerberos SSO as you use for the directory authentication in the scenario above.

You must create service principal names (SPN) in Active Directory for the Tableau Server service.

Use the batch file in the next section to create the SPNs and the keytab file.

After you have created the SPNs, upload the keytab file as described in Configure Kerberos.

Batch file: Set SPN and create keytab in Active Directory

You can use a batch file to set the service principal names (SPN) and create a keytab file. These operations are a part of the process to enable Kerberos SSO for Tableau Server (on Windows or Linux) running in Active Directory.

In previous versions of Tableau Server (before 2018.2), the configuration script was generated from the Tableau Server Configuration utility.

To generate a configuration script, copy and paste the following batch file contents into a text file. The batch file creates service principal names (SPN) for Tableau Server and will create a keytab file for the user you specify in the file.

Follow the directions in the file contents. After you have finished customizing the file, save it as a .bat file.

This file must be run in an Active Directory domain by a Domain admin, who will be prompted for the service account password of the account you specify in the file.

The batch file uses the Windows set, setspn, and ktpass commands.
Note: The batch file below is self-documented. However, if you do not have experience with Kerberos and generating keytab files, we recommend that you read the Microsoft blog post, All you need to know about Keytab files, before proceeding. Environmental details in your organization may require additional configuration of the ktpass command. For example, you must determine what to set for the /crypto parameter. We recommend specifying a single /crypto value that is required by your KDC. See the Microsoft article, ktpass for the full list of supported values for the /crypto parameter.

Creating a keytab file for user authentication in Active Directory must be performed on a Windows computer as specified here. Creating this keytab file on a Linux computer is not supported.

SPN and keytab batch file contents

@echo off
setlocal EnableDelayedExpansion

REM ******

REM This script generates the Service Principal Names (SPNs) and keytab files required for
REM Kerberos SSO with Apache.
REM This script executes set, setspn, and ktpass commands included in any Windows Server
REM Operating System from 2003 on.
REM Before running this script you must enter configuration information for the setspn and
REM ktpass commands.
REM Elements that require your configuration information are enclosed in as such:
REM    ! -- and --!. 
After you customize this file, save it as a .bat file, and run on a domain-joined computer.
REM This script must be run by a Domain admin.

REM **********

REM The following set command will prompt the domain admin for credentials of the Tableau Server service account. This account must be a valid domain user account. If the password contains a literal " (blackslash - double quote), all backslashes immediately before the double quote must be duplicated when typed for the password to work, e.g. if password contains " replace with \, if passwords contains \ replace with \\

set /p adpass= "Enter password for the Tableau Server service account."
set adpass=!adpass:"="!

REM **********

REM The following setspn commands create the SPN in the domain. More information on setspn can be found here:
REM Enter the canonical FQDN and the host names for Tableau Server followed by the Tableau Server service account name.
REM Use this syntax: HTTP/hostname domain\service_account_name.
REM The example below shows syntax for a computer named
"tableau01" in the "example.lan"
REM domain, with service account, "tab-serv-account":
REM setspn -s HTTP/tableau01 example\tab-serv-account
REM setspn -s HTTP/tableau01.example.lan example\tab-serv-account
REM DNS and AD are not case sensitive, but the keytab files are. Verify that host names
REM match letter case as stored in DNS.
REM Use Windows Server's DNS Manager utility to verify host name case.

REM **********

echo Creating SPNs...
setspn -s HTTP/!-replace with canonical host name and service account -!
setspn -s HTTP/!-replace with canonical FQDN and service account -!

REM **********

REM The following commands create the keytab file in the same directory where the
REM bat file is run. More information on ktpass can be found here:
REM https://docs.microsoft.com/en-us/windows-server-/administration/windows-commands/ktpass
REM Note: keytab files are case-sensitive.
REM The realm following the FQDN should be all uppercase.
REM Syntax is:
REM ktpass /princ HTTP/!-FQDN-!@!-Kerberos_Realm-! /pass !adpass!
REM /pttype KRB5_NTPRINCIPAL /crypto !-cipher-! /out keyt-abs\kerberos.keytab
Tableau Server on Linux Administrator Guide

REM Best practice: specify the /crypto value that is required by your KDC.
REM Specifying /crypto All will result in passwords stored with RC4 cipher, which is
REM no longer considered secure.
REM The following example shows the ktpass syntax with the example.lan configuration from above:
REM ktpass /princ HTTP/tableau01.example.lan@EXAMPLE.LAN /pass !adpass! /ptype KRB5_NT_PRINCIPAL /crypto AES256-SHA1 /out keyt-abs\kerberos.keytab

REM ***********

echo Creating Keytab files in %CD%\keytabs
mkdir keytabs
ktpass /princ HTTP/!-FQDN-@!-Kerberos_REALM-! /pass !adpass! /ptype KRB5_NT_PRINCIPAL /crypto AES256-SHA1 /out keyt-abs\kerberos.keytab

Operating system

If your organization uses Kerberos for authentication, then the computer where Tableau Server is running must be authenticated with the Kerberos realm in which it’s running.

<table>
<thead>
<tr>
<th>Tableau is running on...</th>
<th>Need to manually generate a keytab?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Windows in Active Directory domain</td>
<td>No</td>
</tr>
<tr>
<td>Linux in Active Directory domain</td>
<td>Yes</td>
</tr>
<tr>
<td>Windows or Linux in non-Active Directory environment</td>
<td>Yes</td>
</tr>
</tbody>
</table>

If you are running Tableau Server on Windows, and the computer is joined to the Active Directory, then you do not need to manage or generate a keytab file for the operating system.
If you are running Tableau Server on Linux in a Kerberos realm (MIT KDC or Active Directory), then you will need to generate a keytab file specifically for the computer operating system. The keytab you create for the computer should be specifically for OS authentication. Do not use the same keytab file for OS authentication that you will be using for the other services described later in this topic.

Directory service

If your organization uses a directory service, such as LDAP or Active Directory, to manage user identity, then Tableau Server requires read-only access to the directory.

Alternatively, you can configure Tableau Server to manage all accounts by installing with a local identity store. In this case, you do not need a keytab.

The following table summarizes keytab requirements:

<table>
<thead>
<tr>
<th>Tableau is running on...</th>
<th>Directory service</th>
<th>Need to manually generate a keytab?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Windows in AD domain</td>
<td>Active Directory</td>
<td>No</td>
</tr>
<tr>
<td>Windows</td>
<td>LDAP (GSSAPI bind)</td>
<td>Yes</td>
</tr>
<tr>
<td>Linux</td>
<td>Active Directory or LDAP (GSSAPI bind)</td>
<td>Yes</td>
</tr>
<tr>
<td>Windows or Linux</td>
<td>Active Directory or LDAP (Simple bind)</td>
<td>No</td>
</tr>
<tr>
<td>Windows or Linux</td>
<td>Local identity store</td>
<td>No keytab required.</td>
</tr>
</tbody>
</table>

If you need to manually generate a keytab for this scenario, then you will use it for GSSAPI bind to the directory. Follow these recommendations:

- Create a service account in your directory for Tableau Server.
- Create a keytab specifically for the Tableau Server service account. Do not reuse the
keytab file that the computer account/OS uses to authenticate.

- Upload the keytab file as part of the json configuration of the Tableau Server identity store. See identityStore Entity.

Datasource delegation

You can also use Kerberos delegation to access data sources in an Active Directory. In this scenario, users can be authenticated to Tableau Server with any supported authentication mechanism (SAML, local authentication, Kerberos, etc), but can access datasources that are enabled by Kerberos.

<table>
<thead>
<tr>
<th>Tableau is running on...</th>
<th>Need to manually generate a keytab?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Windows in Active Directory domain</td>
<td>Yes</td>
</tr>
<tr>
<td>Linux in Active Directory domain</td>
<td>Yes</td>
</tr>
<tr>
<td>Windows or Linux in non-Active Directory environment</td>
<td>Not a supported scenario.</td>
</tr>
</tbody>
</table>

Follow these recommendations:

- The keytab file that you use for Kerberos delegation can be the same keytab that you use for Kerberos user authentication (SSO).

- The keytab must be mapped to the service principal for Kerberos delegation in Active Directory.

- You may use the same keytab for multiple data sources.

For more information, see the following configuration topics:

- Tableau Server on Linux: [Enable Kerberos Delegation](#)
- Tableau Server on Windows: [Enabling Kerberos Delegation](#)
Configure Kerberos

You can configure Tableau Server to use Kerberos. This allows you to provide a single sign-on (SSO) experience across all the applications in your organization. Before you configure Tableau Server for Kerberos make sure your environment meets the Kerberos Requirements.

Note: Kerberos constrained delegation for SSO to Tableau Server is not supported. (Constrained delegation for data sources is supported.) For more information, see Single-Sign On (SSO) in Kerberos Requirements.

To configure Kerberos, you must first enable Kerberos, and then specify a keytab file for user authentication. The keytab file you specify must be configured with the service provider name for the Tableau Server for user authentication. If you are using Kerberos authentication for data sources, those credentials should be included in the single keytab file that you will specify during Kerberos configuration on Tableau Server.

Use the TSM web interface

1. Open TSM in a browser:

   https://<tsm-computer-name>:8850. For more information, see Sign in to Tableau Services Manager Web UI.

2. Click User Identity & Access on the Configuration tab and then click Authentication Method.


5. To copy the keytab file to the server, click Select File, and then browse to the file on your computer.
6. Click **Save Pending Changes** after you've entered your configuration information.

7. Click **Pending Changes** at the top of the page:

8. Click **Apply Changes and Restart**.

Use the TSM CLI

1. Copy the keytab file to the computer running Tableau Server and run the following command to set permissions on the file:

   ```bash
   chmod 644 "/path/keytab_file"
   ```

   If you are running Tableau Server on in a distributed cluster deployment, then you will need to manually distribute the keytab file to each node and then set the permissions. Copy the keytab file to the same directory on each node in the cluster. After you have copied the keytab file to each node and set permissions on the file, then run the following TSM commands on one node. The configuration will propagate to each node.

2. Type the following command to specify the location and name of the keytab file:
3. Type the following command to enable Kerberos:

```bash
tsm authentication kerberos enable
```

4. Run `tsm pending-changes apply` to apply changes.

   The `pending-changes apply` command displays a prompt to let you know this will restart Tableau Server if the server is running. The prompt displays even if the server is stopped, but in that case there is no restart. You can suppress the prompt using the `--ignore-prompt` option, but this does not change the restart behavior. For more information, see `tsm pending-changes apply`.

Confirm your SSO configuration

Once Tableau Server has restarted, test your Kerberos configuration from a web browser on a different computer by typing the Tableau Server name in the URL window:

![Tableau Server URL](image)

You should be automatically authenticated to Tableau Server.

Tableau Client Support for Kerberos SSO

This article describes some requirements for and nuances with using Kerberos single sign-on (SSO) with Tableau Server, depending on the particular Tableau client and operating
Tableau Server on Linux Administrator Guide

system. Tableau clients covered in this article include common web browsers, Tableau Desktop, and the Tableau Mobile app.

General browser client support

To use browser-based Kerberos Single Sign-on (SSO), the following must be true:

- Kerberos must be enabled on Tableau Server.
- The user must have a user name and password to sign in to Tableau Server.

**Note:** If Kerberos SSO fails, users can fall back on their user name and passwords credentials.

- The user must be authenticated to Active Directory through Kerberos on the client computer or mobile device. Specifically, this means that they have a Kerberos Ticket Granting Ticket (TGT).

Tableau Desktop and browser clients

On Windows or the Mac, you can use Kerberos SSO to sign in to Tableau Server from the following versions of Tableau Desktop or browser. Where noted, additional configuration is required.

Windows

- Tableau Desktop 8.3 or later supported.
- Internet Explorer - supported, may require configuration - see **Note 1**.
- Chrome - supported, may require configuration - see **Note 1**.
- Firefox - requires configuration - see **Note 2**.
- Safari - not supported.

Mac OS X
Tableau Desktop 8.3 or newer
Safari - supported
Chrome - see Note 3
Firefox - see Note 2
Internet Explorer - not supported

Tableau Mobile app clients

On a Mac iOS or Android device, you can use the following Tableau Mobile or mobile browser versions to use Kerberos authentication to Tableau Server:

Mac iOS

- Tableau App 8.3 or newer - see Note 4
- Safari - see Note 4
- Chrome - not supported

Android - see Note 5

- Tableau App 8.3 or newer
- Android Browser
- Chrome

OS- and browser-specific notes

The following notes describe configuration requirements or issues with specific operating system and client combinations.

**Note 1: Internet Explorer or Chrome on Windows desktop**

Kerberos SSO is supported in both Internet Explorer and Chrome, but it requires configuration in **Windows Internet Options**:

1. Enable **Integrated Windows Authentication**.

2. Verify that Tableau Server URL is in the local intranet zone.
Internet Explorer can sometimes detect intranet zones and configure this setting. If it has not detected and configured the Tableau Server URL, you must manually add the URL to the local intranet zone.

To enable Integrated Windows Authentication:

1. In Windows Control Panel, open Internet Options.
2. On the Advanced tab scroll down to the Security section.
4. Click Apply.

To verify or add the Tableau Server URL to the local intranet zone:

1. In Windows Control Panel, open Internet Options.
2. On the Security tab, select Local intranet, and then click Sites.
3. On the Local intranet dialog box, click Advanced.

In the Websites field, look for the internal Tableau Server URL.

In some organizations, IT administrators will use a wildcard (*) to specify internal URLs. For example, the following URL includes all servers in the internal example.lan namespace in the local intranet zone:

https://*.example.lan

The following image shows a specific URL of https://tableau.example.lan.
4. If the Tableau Server URL or a wildcard URL is not specified in the **Websites** field, enter the Tableau Server URL in the **Add the website to the zone** field, click **Add**, and then click **OK**.

   If the Tableau Server URL is already listed in **Websites**, you can simply close the dialog.

**Note 2: Firefox on Windows or Mac OS X desktop**

You can use Firefox with Kerberos SSO on either Windows or Mac to sign in to Tableau Server. To do this, you must complete the following steps to configure Firefox to support Kerberos:

1. In Firefox, enter `about:config` in the address bar.

2. Click **I'll be careful, I promise** when warned about changing advanced settings.

3. Enter `negotiate` in the **Search** box.
4. Double-click `network.negotiate-auth.allow-non-fqdn`, and then set the value to `true`.

5. Double-click `network.negotiate-auth.trusted-uris` and enter the Tableau Server fully qualified domain name (FQDN). For example, `tableau.example.com`.

**Note 3: Chrome on Mac OS X desktop**

According to Chrome documentation, Kerberos SSO works on a Mac when you launch Chrome from a terminal window with the following command:

```
open -a "Google Chrome.app" --args --auth-server-whitelist="tableauserver.example.com"
```

where `tableauserver.example.com` is the URL for Tableau Server in your environment.

However, we have found inconsistent results in our testing. Therefore, if you want to use Kerberos SSO on a Mac, we recommend that you use Safari or Firefox. For more information, see the *Integrated Authentication* section at [HTTP authentication](https://www.chromium.org) on The Chromium Projects site.
Note: Users can still use Chrome on Mac OS X to sign in to Tableau Server, but they might be prompted to enter their user name and password (single sign-on may not work).

Note 4: Mobile Safari or Tableau Mobile on Mac iOS

Kerberos SSO is supported if iOS is configured for Kerberos. The iOS device must have a Kerberos authentication configuration profile installed. This is usually done by an enterprise IT group. Tableau Support cannot assist with configuring iOS devices for Kerberos.

Note 5: Android platform

Kerberos SSO is not supported on the Android operating system because there is no platform-level support for Kerberos. You can still use your Android device and the Tableau Mobile app or a supported mobile browser to connect to and sign in to Tableau Server.

Troubleshoot Kerberos

The troubleshooting suggestions in this topic are divided into issues related to single sign-on (SSO) on the server and issues with the delegated data sources.

Single sign-on to Tableau Server

In a Kerberos SSO environment, a user signing in to Tableau Server from a web browser or Tableau Desktop might see a message indicating that Tableau Server can't sign them in automatically (using single sign-on). It suggests that they provide a Tableau Server user name and password instead.
Troubleshooting sign-in errors on the client computer

• **Enter the user name and password**—To check the user’s general access to Tableau Server, sign in by entering the user’s name and password.

  If these credentials fail, the user might not be a user on Tableau Server. For Kerberos SSO to work, the user must be able to access Tableau Server, and they must be granted a Ticket Granting Ticket (TGT) by Active Directory (as described in the **TGT** item later in this list).

• **Check other users’ SSO credentials**—Try to connect with SSO to Tableau Server using other user accounts. If all users are affected, the problem might be in the Kerberos configuration.

• **Use a computer other than the server computer**—Kerberos SSO does not work when you sign in to Tableau Server on localhost. Clients must connect from a computer other than the Tableau Server computer.

• **Use a server name, not IP address**—Kerberos SSO does not work if you enter an
IP address as the Tableau Server name. In addition, the server name you use to access Tableau Server must match the name used in the Kerberos configuration (see Key table entry, below).

- **Confirm that the client has TGT**—The client computer must have a TGT (Ticket Granting Ticket) from the Active Directory domain. Constrained delegation, with the proxy granting a ticket, is not supported.

  To confirm the client computer has a TGT, do the following:
  
  - On Windows, open a command prompt and type the following: `klist tgt`
  - On the Mac, open a terminal window and type the following: `klist`

  The output should show a TGT for the user/domain trying to authenticate to Tableau Server.

  The client computer might not have a TGT in the following circumstances:
  
  - The client computer is using a VPN connection.
  - The client computer is not joined to the domain (for example, it is a non-work computer being used at work).
  - The user signed in to the computer with a local (non-domain) account.
  - The computer is a Mac that is not using Active Directory as a network account server.

- **Confirm browser version and settings**—For web browser sign-in, make sure the browser is supported for Kerberos and, if necessary, is configured correctly.

  - Internet Explorer (IE) and Chrome work “out of the box” on Windows.
  - Safari works “out of the box” on the Mac.
  - Firefox requires additional configuration.
Troubleshooting sign-in errors on the server

If you cannot solve the problem from the client computer, your next steps are to troubleshoot on the computer running Tableau Server. The administrator can use the request ID to locate the sign-in attempt in the Apache logs on Tableau Server.

- **Log files**—Check the Apache error.log for an error with the exact time/date of the failed sign-in attempt.

- **Key table entry**—If the error.log entry includes the message, “No key table entry matching HTTP/<servername>.<domain>.<org>@”, for example:

  [Fri Oct 24 10:58:46.087683 2014] [:error] [pid 2104:tid 4776] [client 10.10.1.62:56789] gss_acquire_credential() failed: Unspecified GSS failure. Minor code may provide more information (, No key table entry found matching HTTP/server-name.domain.com@)

  This error is a result of a mismatch of any of the following:

  - **Tableau Server URL** - The URL used by the client computer to access the server.

    This is the name that you type into Tableau Desktop or a browser address bar. It could be a shortname (http://servername) or a fully-qualified domain name (http://servername.domain.com)
- **DNS reverse lookup** for the server IP address.

  This looks up a DNS name using an IP address.

  At a command prompt type:

  ```
  ping servername
  ```

  with the IP address returned by pinging the server, do a reverse DNS lookup type:

  ```
  nslookup <ip address>
  ```

  The nslookup command will return network information for the IP address. In the *Non-authoritative answer* portion of the response, verify that the fully qualified domain name (FQDN) matches the following configured values:

  - The Kerberos .keytab file
  - Service Principal Name (SPN) for the server

  For more information about configuring these values, see Understanding Keytab Requirements.

### Verify Kerberos configuration script

You may need to modify the ktpass command that you used to generate the keytab file for environmental variables. Review the troubleshooting steps in the Knowledge Base article, [Unable to Generate Kerberos Script Configuration for Tableau Server](#).

### Data source SSO

#### Delegated data source access failures

Check the vizqlserver log files for "workgroup-auth-mode."

Look for "workgroup-auth-mode" in the log files. It should say "kerberos-impersonate" not "as-is".
Kerberos delegation multi-domain configuration

Tableau Server can delegate users from other Active Directory domains. If your database uses MIT Kerberos, you may need to adjust your Kerberos principal to database user mapping. Specifically, you will need to update krb5.conf with rules for each Kerberos realm that users will connect from. Use the auth_to_local tag in the [realms] section to map principal names to local user names.

For example, consider a user, EXAMPLE\jsmith, whose Kerberos Principal is jsmith@EXAMPLE.LAN. In this case, Tableau Server will specify a delegated user, jsmith@EXAMPLE. Tableau Server will use the Active Directory legacy domain alias as the Kerberos Realm.

The target database may already have a rule such as the following to map the user, jsmith@EXAMPLE.LAN to the database user, jsmith.

```
EXAMPLE.LAN = {
    RULE: [1:$1@$0](.*@EXAMPLE.LAN)@.*//@
    DEFAULT
}
```

To support delegation, you must add another rule to map jsmith@EXAMPLE to a database user:

```
EXAMPLE.LAN = {
    RULE: [1:$1@$0](.*@EXAMPLE.LAN)@.*//@
    RULE: [1:$1@$0](.*@EXAMPLE)s/@.*//@
    DEFAULT
}
```

See the MIT Kerberos Documentation topic, krb5.conf, for more information.
Cross-domain constrained delegation

In some cross-domain scenarios where the KDC is running on a Windows Server prior to Windows 2012, delegation may fail. Errors you may see include:

- SQL Server Network Interfaces: The system cannot contact a domain controller to service the authentication request. Please try again later.
- SQL Server Native Client: Cannot generate SSPI context
- The Domain Controller returns: KRB-ERR-POLICY error with a status STATUS_CROSSREALM_DELEGATION_FAILURE (0xc000040b).

Cross-domain refers to a scenario where Tableau Server is running in a different domain than the data source with different service accounts. For example:

- Tableau Server runs on DomainA with DomainA service account.
- SQL Server runs on DomainB with DomainB service account.

Traditional constrained delegation only works if both servers are in the same domain. The user can come from other domains.

If you are seeing the errors noted above, then to enable this scenario, your Active Directory administrator should remove any traditional constrained delegation which is configured on the delegating account. Removing delegation can be achieved with Active Directory management tools or by removing the values associated with the Active Directory property, msDS-AllowedToDelegateTo.

If you want to preserve an existing single domain delegation alongside cross-domain delegation, you must configure both using resource-based constrained delegation.

For more information about Kerberos and constrained delegation, see the Microsoft topic, Kerberos Constrained Delegation Overview.

Configure Mutual SSL Authentication

Using mutual SSL, you can provide users of Tableau Desktop and other approved Tableau clients a secure, direct-access experience to Tableau Server. With mutual SSL, when a client with a valid SSL certificate connects to Tableau Server, Tableau Server confirms the
existence of the client certificate and authenticates the user, based on the user name in the client certificate. If the client does not have a valid SSL certificate, Tableau Server can refuse the connection. You can also configure Tableau Server to fall back to username/password authentication if mutual SSL fails.

Mutual SSL authentication is not supported on Tableau Mobile.

Use the TSM web interface

1. Configure SSL for External HTTP Traffic to and from Tableau Server.

2. Open TSM in a browser:
   
   https://<tsm-computer-name>:8850. For more information, see Sign in to Tableau Services Manager Web UI.


4. Under Authentication Method, select Mutual SSL in the drop-down menu.

5. Under Mutual SSL, select Use mutual SSL and automatic sign in with client certificates.

6. Click Select File and upload your certificate authority (CA) issued certificate to the server.
   
   This certificate must be a valid PEM-encoded x509 certificate with the extension .crt.

7. Enter remaining SSL configuration information for your organization.

   **Username format:** When Tableau Server is configured for mutual SSL, the server gets the user name from the client certificate, so it can establish a direct sign-in for the client user. The name that Tableau Server uses depends on how Tableau Server is configured for user authentication:

   - Local Authentication—Tableau Server uses the UPN (User Principal Name) from the certificate.
Active Directory (AD)—Tableau Server uses LDAP (Lightweight Directory Access Protocol) to get the user name.

Alternatively, you can set Tableau Server to use the CN (Common Name) from the client certificate.

8. Click **Save Pending Changes** after you've entered your configuration information.

9. Click **Pending Changes** at the top of the page:

10. Click **Apply Changes and Restart**.

Use the TSM CLI

**Step 1: Require SSL for external server communication**

To configure Tableau Server to use SSL for external communication between Tableau Server and web clients, run the `external-ssl enable` command as follows, providing the names for the server certificate’s .crt and .key files:
Tableau Server on Linux Administrator Guide

tsm security external-ssl enable --cert-file <file.crt> --key-file <file.key>

- For --cert-file and --key-file, specify the location and file name where you saved the server’s CA-issued SSL certificate (.crt) and key (.key) files.

- The above command assumes the you are signed in as a user that has the Server Administrator site role on Tableau Server. You can instead use the -u and -p parameters to specify an administrator user and password.

- If the certificate key file requires a passphrase, include the --passphrase parameter and value.

Step 2: Use mutual SSL

Add mutual authentication between the server and each client, and allow for Tableau client users to be authenticated directly after the first time they provide their credentials.

1. Run the following command:

   tsm authentication mutual-ssl configure --cert-file <file.crt>

   For --cert-file, specify the location and file name of the server’s CA certificate (.crt) file, as in the previous step for external SSL.

   See the remaining sections in this article for any additional options you might want to include with the mutual-ssl configure command.

2. Run the following commands to enable mutual SSL and apply the changes:

   tsm authentication mutual-ssl enable

   tsm pending-changes apply

   The pending-changes apply command displays a prompt to let you know this will restart Tableau Server if the server is running. The prompt displays even if the

Tableau Software  
Version: 2019.4  
795
server is stopped, but in that case there is no restart. You can suppress the prompt using the --ignore-prompt option, but this does not change the restart behavior. For more information, see tsm pending-changes apply.

Additional options for mutual SSL

You can use mutual-ssl configure to configure Tableau Server to support the following options.

For more information, see tsm authentication mutual-ssl <commands>.

Fallback authentication

When Tableau Server is configured for mutual SSL, authentication is automatic and clients must have a valid certificate. You can configure Tableau Server to allow a fallback option, to accept user name and password authentication.

```bash
    tsm authentication mutual-ssl configure -fb true
```

Tableau Server accepts username and password authentication from REST API clients, even if the above option is set to false.

User name mapping

When Tableau Server is configured for mutual SSL, the server authenticates the user directly by getting the user name from their client certificate. The name that Tableau Server uses depends on how the server is configured for user authentication:

- **Local Authentication**—uses the UPN (User Principal Name) from the certificate.

- **Active Directory (AD)**—uses LDAP (Lightweight Directory Access Protocol) to get the user name.

You can override either of these defaults to set Tableau Server to use the common name.

```bash
    tsm authentication mutual-ssl configure -m cn
```
Certificate Revocation List (CRL)

You might need to specify a CRL if you suspect that a private key has been compromised, or if a certificate authority (CA) did not issue a certificate properly.

tsm authentication mutual-ssl configure -rf <revoke-file.pem>

Mapping a Client Certificate to a User During Mutual Authentication

When you use mutual (two-way) SSL authentication, the client presents its certificate to Tableau Server as part of the authentication process. Tableau Server then maps user information in the client certificate to a known user identity. The strategy that Tableau Server uses to perform client mapping depends on the content of your organization’s client certificates.

This topic discusses the ways information in a client certificate can map to a user identity and how to change the way Tableau Server performs that mapping. To understand how the mapping happens and whether you need to change it, you must know how client certificates are structured in your organization.

- **User-name mapping options**
- **Change the certificate mapping**
- **Address user-name ambiguity in multi-domain organizations**

User-name mapping options

Tableau Server uses one of the following approaches to map a client certificate to a user identity:

- **Active Directory.** If Tableau Server is configured to use Active Directory for user authentication, when Tableau Server receives a client certificate, it passes the certificate to Active Directory, which maps the certificate to an Active Directory identity.
Any explicit user name information in the certificate is ignored.

**Note:** This approach requires client certificates to be published for the user accounts in Active Directory.

- **User principal name (UPN).** A client certificate can be configured to store the user name in the user principal name field. Tableau Server reads the UPN value and maps it to a user in Active Directory or to a local user.

- **Common name (CN).** A client certificate can be configured to store the user name in the common name field of the certificate. Tableau Server reads the CN value and maps it to a user in Active Directory or to a local user.

If you configure the server for Active Directory authentication and UPN or CN user-name mapping, put the user name in one of the following formats:

username, domain/username, or username@domain.

For example: jsmith, example.org/jsmith, or jsmith@example.org.

If the server uses local authentication, the format of the name in the UPN or CN fields is not predetermined, but the name in the field must match a user name on the server.

Change the certificate mapping

You use the tsm authentication mutual-ssl <commands> commands to map a client certificate to a user identity in Tableau Server:

```
rs authentication mutual-ssl configure -m <value>
```

Possible values are ldap for Active Directory mapping, upn for UPN mapping, or cn for CN mapping.

When you first install and configure Tableau Server, the server sets the default user-name mapping to match the server’s authentication type:
If the server is configured to use Active Directory, it also uses Active Directory for mapping the certificate to the user identity.

If the server is configured to use local authentication, the server gets the user-name value from the UPN field in the certificate.

If the default behavior for how Tableau Server maps a user name to an identity is not correct for your server configuration, run the following set of commands to change the mapping to use the CN value:

```
tsm authentication mutual-ssl configure -m cn
```

```
tsm pending-changes apply
```

The `pending-changes apply` command displays a prompt to let you know this will restart Tableau Server if the server is running. The prompt displays even if the server is stopped, but in that case there is no restart. You can suppress the prompt using the `--ignore-prompt` option, but this does not change the restart behavior. For more information, see `tsm pending-changes apply`.

Address user-name mapping ambiguity in multi-domain organizations

Under some circumstances, the user name in a certificate’s UPN or CN field can be ambiguous. This ambiguity can lead to unexpected results when the user name is mapped to a user identity on the server.

For example, if Tableau Server is presented with a user name that does not include a domain, the server maps the user name to an identity using the default domain. This can cause an incorrect user-name mapping, potentially assigning a user a different user’s identity and permissions.

This can occur particularly in environments where the following conditions apply:

* Your organization supports multiple Active Directory domains.
* The server is configured to use Active Directory authentication.
The server is configured to use UPN or CN mapping.

Some users have the same user name but different domains. For example, jsmith@example.org and jsmith@example.com.

The user name in the certificate’s UPN or CN fields does not include the domain as part of the user name—for example, it shows jsmith.

To avoid incorrect user-name mapping, make sure the client certificates include fully qualified user names with the domain, using the format jsmith@example.org or example.org/jsmith.

OpenID Connect

You can configure Tableau Server to support OpenID Connect for single sign-in (SSO). OpenID Connect is a standard authentication protocol that lets users sign in to an identity provider (IdP) such as Google. After they’ve successfully signed in to their IdP, they are automatically signed in to Tableau Server.

Configuring OpenID Connect involves several steps. The topics in this section provide general information about using Tableau Server with OpenID Connect, and provide a sequence for configuring the IdP and Tableau Server.

Authentication overview

This section describes the OpenID Connect authentication process with Tableau Server.
Step 1: A user attempts to log in to Tableau Server from a client computer.

Step 2: Tableau Server redirects the request for authentication to the IdP gateway.

Step 3: The user is prompted for credentials and successfully authenticates to the IdP. The IdP responds with a redirect URL back to Tableau Server. The redirect URL includes an authorization code for the user.

Step 4: The client is redirected to Tableau Server and presents the authorization code.

Step 5: Tableau Server presents the client's authorization code to the IdP along with its own client credentials. Tableau Server is also client of the IdP. This step is intended to prevent spoofing or man-in-the-middle attacks.

Step 6: The IdP returns an access token and an ID token to Tableau Server.
• **JSON Web Token (JWT) validation:** By default Tableau Server performs a validation of the IdP JWT. During discovery, Tableau Server retrieves the public keys specified by the `jwks_uri` in the IdP configuration discovery document. Tableau Server validates the ID token for expiry and then verifies the JSON web signature (JWS), the issuer (IdP), and the client ID. You can learn more about the JWT process in the OpenID documentation, 10. Signatures and Encryption, and the IETF proposed standard, JSON Web Token. We recommend leaving JWT validation enabled, unless your Idp does not support it.

• The ID token is a set of attribute key-pairs for the user. The key-pairs are called **claims.** Here is an example IdP claim for a user:

```
  "sub" : "7gYhRR3HiRRCaRcgvY50ubrtjGQBMJW4rXbpPFp-g2cptHP62m2sqowM7G1LwjN5",
  "email" : "alice@tableau.com",
  "email_verified" : true,
  "name" : "Alice Adams",
  "given_name" : "Alice",
  "family_name" : "Adams",
  "phone_number" : "+359 (99) 100200305",
  "profile" : "https://tableau.com/users/alice"
```

**Step 7: Tableau Server identifies the user from the IdP claims and completes the authentication request from Step 1.** Tableau Server searches for the user’s account record stored in the repository by matching the "sub" (subject identifier) to identify the correct user account. If no user account is stored with the sub claim value, then Tableau Server searches for a username in the repository that matches the "email" claim from the IdP. When a username match succeeds, Tableau Server will store the corresponding sub claim to the user’s record in the repository. Tableau Server can be configured to use different claims for this process. See Requirements for Using OpenID Connect.

**Step 8: Tableau Server authorizes the user.**
Tableau Server on Linux Administrator Guide

How Tableau Server Works with OpenID Connect

OpenID Connect is a flexible protocol that supports many options for the information that’s exchanged between a service provider (here, Tableau Server) and an IdP. The following list provides details about the Tableau Server implementation of OpenID Connect. These details can help you understand what types of information Tableau Server sends and expects, and how to configure an IdP.

- Tableau Server supports only the OpenID Authorization Code Flow as described in the OpenID Connect final specification.

- Tableau Server relies on using discovery or a provider URL to retrieve the OpenID Provider metadata. Alternatively, you can host a static discovery document on Tableau Server. For more information see Configure Tableau Server for OpenID Connect.

- Tableau Server supports only the client_secret_jwt Client Authentication method specified in the OpenID Connect specification. In addition, Tableau Server supports only RSA Asymmetric Encryption for handling the JWT. However, you can turn off JWT validation. See tsm authentication openid <commands>.

- Tableau Server expects a kid value in the id_token attribute’s JOSE Header. This value is matched with one of the keys found in the JWK Set document, whose URI is specified by the jwks_uri value in the OpenID discovery document. A kid value must be present even if there is only one key in the JWK Set document.

- Tableau Server does include OpenID support for the JWK x5c parameter or for using X.509 certificates.

Requirements for Using OpenID Connect

This topic describes the requirements to use OpenID Connect with Tableau Server.

IdP account

You must have access to an identity provider (IdP) that supports the OpenID Connect (OIDC) protocol. You must also have an account with the IdP. OpenID Connect is supported
by many identity providers. The OIDC protocol is an open and flexible standard, and as such, not all implementations of the standard are identical. As you configure Tableau Server for OIDC, work with your IdP.

The Google IdP implementation has been extensively tested with Tableau Server and is the model IdP for the configuration documented in these topics.

Local authentication

To use OpenID Connect on Tableau Server, the server must be configured to use local authentication—that is, the server must be configured so that you explicitly create users on the Tableau Server, rather than importing them from Active Directory. Active Directory authentication is not supported with OpenID.

IdP claims: mapping users

To sign in successfully to Tableau Server, a given user must be provisioned in OpenID and then mapped to a user account on Tableau Server. OpenID uses a method that relies on claims to share user account attributes with other applications. Claims include user account attributes such as email, phone number, given name, etc. To understand how Tableau Server maps IdP claims to user accounts, see Authentication overview.

Tableau Server relies on the IdP claims to map user accounts from the IdP to those hosted on Tableau Server. By default, Tableau Server expects the IdP to pass the email claim. Depending on your IdP, you may need to configure Tableau Server to use a different IdP claim.

If you are using Google as an IdP, then use the default, email claim to map IdP identities to Tableau Server user accounts. If you are not using Google as an IdP, then work with your IdP to determine the claim for which you should configure Tableau Server.

Default: using email claim to map users

By default, the user’s user name in Tableau Server must match the email claim in the IdP ID token. Therefore, in the default configuration, you must use email addresses (also referred to as UPN) as the username in Tableau Server. If you use Google as the IdP, the
user name in Tableau Server must be the user’s Gmail address (alice@gmail.com). Using a complete email address helps to guarantee the uniqueness of the user name in Tableau Server, even when two users have the same email but are on different email hosts.

**Note:** When you create a user identity in Tableau Server, you specify a user name, password, and optionally an email address. For using OpenID Connect in the default configuration, the user name (expressed as an email address) is the value that must match the user’s name in the IdP. The optional email address in the Tableau Server user identity is not used for OpenID authentication.

**Ignoring the domain name**

You can configure Tableau to ignore the domain portion of an email address when matching the IdP email claim to a user account on Tableau Server. In this scenario, the email claim in the IdP might be alice@example.com, but this will match a user named alice in Tableau Server. Ignoring the domain name might be useful if you already have users defined in Tableau Server that match the user names portion of the email claim, but not the domain portions.

**Important:** We do not recommend ignoring the user domain name without taking precautions. Specifically, verify that user names are unique across the configured domains that you’ve created in your IdP.

Setting Tableau Server to ignore the user domain name has the potential to result in unintended user log on. Consider the case where your IdP has been configured for multiple domains (example.com and tableau.com). If two users with the same first name, but different user accounts (alice@tableau.com and alice@example.com) are in your organization, then the first one to complete the OpenID provisioning sequence will claim the sub mapping in the IdP. If the wrong user is mapped, then the other user will be unable to log on until the associated sub value is reset.
To configure Tableau Server to ignore domain names in user names from the IdP, set 
\texttt{tsm authentication openid configure --ignore-domain to true. For more}
information, see \texttt{tsm authentication openid <commands>}. 

When you change the \texttt{tsm authentication openid configure --ignore-domain} option to ignore the domain in user names, all user names in Tableau Server must have a domain name.

**Using custom claims to map users**

As referenced in Authentication overview, the \texttt{sub} claim is often included in IdP claims. Typically, the \texttt{sub} claim is a unique string that identifies a given user account. The benefit of using a \texttt{sub} claim is that it will not change, even if you or another admin updates other user attributes or IdP claims (email, phone number, etc) associated with that account. By default, Tableau Server identifies and verifies OpenID users according to the \texttt{sub} claim in the IdP ID token.

The OpenID \texttt{sub} claim value must be mapped to the corresponding user in Tableau Server. Since the \texttt{sub} claim is an arbitrary string, a different claim is used to associate accounts during the first sign-in session. The first time a user signs in to Tableau Server with OpenID, Tableau will match the OpenID user account to a corresponding user account on Tableau. By default, Tableau will use the IdP claim, \texttt{email}, to identify the Tableau user. Tableau will then update that user’s record with the \texttt{sub} claim from OpenID. Since the ID token always includes the \texttt{sub} claim along with other claims, on subsequent sessions, Tableau will identify that user with the \texttt{sub} claim only.

For some organizations, mapping user names with the email address is not reliable or not supported by the IdP. Beginning with Tableau Server 10.2, you can map user accounts from any arbitrary IdP claim to the Tableau Server username.

The IdP claim you are using must map exactly to a corresponding Tableau Server username. In the example below, the username is \texttt{kwilliams}. 

To change the IdP claim that is used to map identity on Tableau Server, use the tsm authentication openid map-claims --user-name command. For more information, see tsm authentication openid <commands>.

**Changing the sub claim**

As described above, the sub claim is the identifier that Tableau Server uses to identify users after the initial mapping session. The sub claim is written to the corresponding user account in Tableau Server. If your IdP does not provide a sub claim, then you can specify an arbitrary claim to use instead. Like sub, the claim value you specify must be unique and should not change when other user claims are updated.
To specify a different IdP claim for default sub claim, use the `tsm authentication openid map-claims --id` command. For more information, see `tsm authentication openid <commands>`.

Where `arbitraryClaim` is the name of the IdP claim that you want to use as the replacement for the `sub claim`.

Configure the Identity Provider for OpenID Connect

This topic provides information about configuring an identity provider (IdP) to use OpenID Connect with Tableau Server. This is one step in a multi-step process. The following topics provide information about configuring and using OpenID Connect with Tableau Server.

- OpenID Connect
- Configure the Identity Provider for OpenID Connect (you are here)
- Configure Tableau Server for OpenID Connect
- Signing In to Tableau Server Using OpenID Connect

Configure the IdP

Before you can use OpenID Connect with Tableau Server, you must have an account with an identity provider (IdP) and a project or application with the IdP. When you configure Tableau Server, you will need to be able to provide the following information:

- Provider client ID. This is the identifier that the IdP assigned to your application.
- Provider client secret. This is a token that is used by Tableau to verify the authenticity of the response from the IdP. This value is a secret and should be kept securely.
- Provider configuration URL. This is the URL at the provider’s site that Tableau Server should send authentication requests to.
Redirect URL

Some IdPs will require a redirect URL for your Tableau Server. You can manually construct your URL for the IdP using the following syntax:

<protocol>://<host>/vizportal/api/web/v1/auth/openIdLogin

For example, https://tableau.example.com/vizportal/api/web/v1/auth/openIdLogin.

Example IdP process

The following procedure provides an outline of the steps that you follow with the provider. As an example, the procedure discusses using Google as a provider. However, each provider has a somewhat different flow, so the specifics of the steps (and their order) might vary depending on your provider.

1. Register at the provider’s developer site and sign in. For example, for Google, you can go to the Developers Console at this URL: https://console.developers.google.com

2. Create a new project, application, or relying party account.

3. In the developer dashboard, follow the steps for getting an OAuth 2.0 client ID and client secret. Record these values for later.

   **Note:** Keep the client secret in a secure place.

4. On the developer site, find the URL of the endpoint that the IdP uses for OpenID Connect discovery. For example, Google uses the URL https://accounts.google.com/.well-known/openid-configuration. Record this URL for later.

   Alternatively, if your IdP has provided you with a static discovery document, copy that file to a local directory on the Tableau Server for later.
Configure Tableau Server for OpenID Connect

This topic describes how to configure Tableau Server to use OpenID Connect for single-sign on (SSO). This is one step in a multi-step process. The following topics provide information about configuring and using OpenID Connect with Tableau Server.

- OpenID Connect
- Configure the Identity Provider for OpenID Connect
- Configure Tableau Server for OpenID Connect (you are here)
- Signing In to Tableau Server Using OpenID Connect

**Note:** Before you perform the steps described here, you must configure the OpenID identity provider (IdP) as described in Configure the Identity Provider for OpenID Connect.

Use the TSM web interface

1. Open TSM in a browser:
   
   https://<tsm-computer-name>:8850. For more information, see Sign in to Tableau Services Manager Web UI.

2. Click User **Identity & Access** on the **Configuration** tab and then click **Authentication Method**.


4. Under OpenID Connect, select **Enable OpenID authentication for the server**.

5. Enter the OpenID configuration information for your organization:
Note: If your provider relies on a configuration file hosted on the local computer (rather than a file hosted at a public URL), you can specify the file with the tsm authentication openid <commands>. Use the --metadata-file <file_path> option to specify a local IdP configuration file.

6. Click **Save Pending Changes** after you've entered your configuration information.

7. Click **Pending Changes** at the top of the page:

8. Click **Apply Changes and Restart**.

Use the TSM CLI

The procedure in this section describes how to use TSM command line interface to configure OpenID Connect. You can also use a configuration file for the initial configuration of OpenID
1. Use the configure command of tsm authentication openid <commands> to set the following required options:

   --client-id <id>: Specifies the provider client ID that your IdP has assigned to your application. For example, “laakjwdlnaoiloadjkwha”.

   --client-secret <secret>: Specifies the provider client secret. This is a token that is used by Tableau to verify the authenticity of the response from the IdP. This value is a secret and should be kept securely. For example, “fwah-fkjaw72123=”.  

   --config-url <url> or --metadata-file <file_path>: Specifies location of provider configuration json file. If the provider hosts a public json discovery file, then use --config-url. Otherwise, specify a path on the local computer and file name for --metadata-file instead.

   --return-url <url>: The URL of your server. This is typically is the public name of your server, such as “http://example.tableau.com”.

For example, run the command:


There are additional, optional configurations that you can set for Open ID Connect using either openIDSettings Entity or tsm authentication openid <commands>. In addition, if you need to configure IdP claim mapping, see Options for openid map-claims.

2. Type the following command to enable Open ID Connect:

tsm authentication openid enable
3. Run `tsm pending-changes apply` to apply changes.

   The `pending-changes apply` command displays a prompt to let you know this will restart Tableau Server if the server is running. The prompt displays even if the server is stopped, but in that case there is no restart. You can suppress the prompt using the `--ignore-prompt` option, but this does not change the restart behavior. For more information, see `tsm pending-changes apply`.

Signing In to Tableau Server Using OpenID Connect

This topic provides information about signing in to Tableau Server using OpenID Connect. The following topics provide information about configuring and using OpenID Connect with Tableau Server.

- OpenID Connect
- Configure the Identity Provider for OpenID Connect
- Configure Tableau Server for OpenID Connect
- Signing In to Tableau Server Using OpenID Connect (you are here)

Signing in using OpenID Connect

After Tableau Server has been configured to use OpenID Connect, users who access the server and aren’t already signed in are redirected to the IdP site, where they are prompted to sign in. Users enter the credentials that they have with the IdP. In many cases, the user is also asked to authorize the IdP to share information with Tableau Server, as in the following example:
When a user signs in using OpenID Connect, the IdP sends a unique user identifier (known in OpenID as the sub value) as part of the information that’s redirected to Tableau Server. This sub value is associated with the user’s Tableau user identity.

Restricting sign-in to server administrators for command-line tools

Command-line tools for working with Tableau Server (tabcmd, TSM, and tableau.com) do not support sign-in using OpenID Connect. When OpenID Connect is enabled for the server, these tools still require sign-in using a Tableau Server username and password.

Even if users normally authenticate using OpenID Connect, each user has a Tableau Server username and password. This means that users could use command-line tools like tabcmd. As a security measure, you can make sure that only server administrators can use command-line tools. To do this, use `tsm configuration set to set wgserv- er.authentication.restricted to true`. When this setting is true, only server administrators can sign in to Tableau Server using a username and password; all other users must sign in to the server using a single sign-on (SSO) option like OpenID Connect.
The effect is that users who are not administrators also cannot then use command-line tools. To make this change, run the following sequence of TSM commands:

```
tsm configuration set -k wgserver.authentication.restricted -v true
tsm pending-changes apply
```

OpenID Connect Authentication Request Parameters

The OpenID authentication request sent from Tableau Server passes information using a limited set of parameters, as listed in this topic. If your OpenID IdP requires parameters that are not in the following list, it is not compatible for use with Tableau Server.

- **scope.** This value specifies a profile that tells the IdP what user information claims to return. This value can be configured by a Tableau Server administrator. The default value is "openid email profile". For more information, see Configure the scope value later in this document.

- **response_type.** OpenID Connect supports multiple flows. This value tells the IdP which flow Tableau Server expects. Tableau supports only the authorization code flow, and the value is always set to "code".

- **client_id.** This value specifies the server’s ID (Provider client ID in the Tableau Server Configuration dialog box), which lets the IdP knows where the request came from. It is provided by the IdP when the service is registered. The value is configurable by a Tableau Server administrator.

- **redirect_uri.** This value specifies the URL that the IdP redirects to after the user has authenticated using OpenID Connect. The URL must include the host and protocol (for example, http://example.tableau.com), but Tableau provides the URL endpoint.

- **nonce.** Tableau Server generates a nonce value to verify that the client that it redirected to matches the entity that comes back from the IdP.
Configure the scope value

The scope value indicates to the IdP the information that Tableau Server requests about the user. By default, Tableau Server sends the value "openid profile email". This indicates that Tableau uses OpenID to authenticate (this part of the scope attribute value must always be included) and that Tableau Server is requesting the user profile and email information during the exchange of the user authorization code.

If this default scope is not appropriate for your scenario, you can have Tableau Server request custom information about the user. To do so, you configure the IdP with a custom profile (for example, something like "tableau-scope"). You can then configure Tableau Server to send the scope request using the name of the custom profile.

To change the scope value that Tableau Server requests, use the following TSM CLI command:

```
tsm authentication openid configure --custom-scope-name custom-scope-name
```

**Note:** Tableau Server always includes "openid" as part of the scope value (even if you don't include it in the custom_scope setting).

Changing IdPs in Tableau Server for OpenID Connect

This topic provides information about changing an identity provider (IdP) if you have configured Tableau Server to use OpenID Connect.

Changing providers

You might decide to change the IdP that Tableau Server is configured to use. To do so, you follow the procedure that you used to configure the first IdP: establish an account, get a customer ID and secret, configure Tableau Server with that information, and provide the IdP with the redirect URL for Tableau Server. For more information, see Configure Tableau Server for OpenID Connect.
However, you also need to perform an additional step: you must clear any user identifiers (sub values) that have already been associated with Tableau Server users. The new IdP will have different sub values for each user, and you must clear the existing ones so that Tableau Server can store a new sub value when the user signs in using the new IdP.

To clear sub values for users, use the `tabcmd reset_openid_sub` command. You can reset (that is, clear) sub values for an individual user, as in the following example:

```
tabcmd reset_openid_sub --username jsmith
```

You can also clear the sub value for all users using this command:

```
tabcmd reset_openid_sub --all
```

Troubleshoot OpenID Connect

Use the following topics to troubleshoot OpenID Connect (OIDC) issues in Tableau Server.

OpenID Connect protocol is supported by many identity providers. The OIDC protocol is an open and flexible standard, and as such, not all implementations of the standard are identical. Most issues that administrators encounter when configuring Tableau Server for OIDC are the result of how different identity providers implement OIDC. If you encounter errors as you set up OIDC with Tableau Server, we recommend that you work with your IdP to resolve them.

Signing In from the Command Line

Even if Tableau Server is configured to use OpenID, it is not used if you sign in to Tableau Server using `tabcmd`, the REST API, or the Tableau Data Extract command line utility (provided with Tableau Desktop).

Login Failed

Login can fail with the following message:
Login failure: Identity Provider authentication successful for user <username from IdP>. Failed to find the user in Tableau Server.

This error typically means that there is a mismatch between a username stored in Tableau Server and the username provided by the IdP. To fix this, make sure that they match. For example, if Jane Smith's username is stored in the IdP as jsmith it must be stored in Tableau Server as jsmith as well.

Error 69: "Unable to Sign In"

An error 69 may be returned when you attempt to sign in to Tableau Server with a web browser and receive an error, "Unable to Sign In. Sign in failed. Contact your Tableau Server administrator." The URL that returns this message is https://example.-com/#/error/signin/69?redirectPath=%2.

If you receive this error, check with your IDP provider to verify if the IdP is expecting client_secret_post instead of client_secret_basic (the Tableau default).

If the IdP is expecting client_secret_post, then you must set the vizportal.openid.client_authentication parameter to client_secret_post.

For example; If you receive this error and you have configured OIDC for the Salesforce IdP, then you must set the vizportal.openid.client_authentication parameter.

See tsm configuration set Options for more information.

OpenID Error Log

OpenID authentication takes place outside Tableau Server, so troubleshooting authentication issues can be difficult. However, sign-in attempts are logged by Tableau Server. You can create a snapshot of log files and use them to troubleshoot problems. For more information, see Server Log File Locations.
Note: To log OpenID-related events, `vizportal.log.level` must be set to debug with `tsm` configuration set options.

Check for OpenID errors in the following files in the unzipped log file snapshot:

`\vizportal\vizportal-<n>.log`

**Trusted Authentication**

When you embed Tableau Server views into webpages, everyone who visits the page must be a licensed user on Tableau Server. When users visit the page they are prompted to sign in to Tableau Server before they can see the view. If you already have a way of authenticating users on the webpage or within your web application, you can avoid this prompt and save your users from having to sign in twice by setting up trusted authentication.

Trusted authentication simply means that you have set up a trusted relationship between Tableau Server and one or more web servers. When Tableau Server receives requests from these trusted web servers it assumes that your web server has handled whatever authentication is necessary.

If your web server uses SSPI (Security Support Provider Interface), you do not need to set up trusted authentication. You can embed views and your users will have secure access to them as long as they are licensed Tableau Server users and members of your Active Directory.

Note: Client browsers must be configured to allow third-party cookies if you want to use trusted authentication with embedded views.

How Trusted Authentication Works

The diagram below describes how trusted authentication works between the client's web browser, your web server(s) and Tableau Server.
User visits the webpage: When a user visits the webpage with the embedded Tableau Server view, the webpage sends a GET request to your web server for the HTML for that page.

Web server POSTS to Tableau Server: The web server sends a POST request to the trusted Tableau Server (for example, https://tabaserver/trusted, not https://tabserver). That POST request must have a username parameter. The username value must be the username for a licensed Tableau Server user. If Tableau Server is hosting multiple sites and the view is on a site other than the Default site, then the POST request must also include a target_site parameter.

Tableau Server creates a ticket: Tableau Server checks the IP address or host name of the web server (192.168.1.XXX in the above diagram) that sent the POST request. If the web server is listed as a trusted host then Tableau Server creates a ticket in the form of a unique string. Tickets must be redeemed within three minutes after they are issued.
Tableau Server responds to the POST request with that ticket. Or if there is an error and the ticket cannot be created, then Tableau Server responds with a value of \(-1\).

4. **Web server passes the URL to the browser:** The web server constructs the URL for the view and inserts it into the HTML for the page. The ticket is included (for example, https://tabserver/trusted/<ticket>/views/requested_view_name). The web server passes the HTML back to the client’s web browser.

5. **Browser requests view from Tableau Server:** The client web browser sends a GET request to Tableau Server that includes the URL with the ticket.

6. **Tableau Server redeems the ticket:** Tableau Server redeems the ticket, creates a session, logs the user in, removes the ticket from the URL, and then sends the final URL for the embedded view to the client.

The session allows the user to access any of the views that the user would have if they logged onto the server. In the default configuration, users authenticated with trusted tickets have restricted access such that only views are available. They cannot access workbooks, project pages, or other content hosted on the server.

To change this behavior, see the `wgserver.unrestricted_ticket` option at `tsm configuration set Options`.

How is a trusted ticket stored?

Tableau Server stores trusted tickets in the Tableau Server repository using the following process:

1. Tableau Server generates a two-part ticket: the first part is a Base64-encoded unique ID (UUID) and the second part is a 24-character random secret string.
2. Tableau Server hashes the secret string and stores it with the unique ID in the repository. Hashing takes the secret string as input, and uses an algorithm to compute a unique string. This unique string protects the security of the secret string from unauthorized users.
3. Tableau Server sends the Base64 UUID and the original 24-character random string to the client.
4. The client returns the Base64 UUID and the original 24-character secret string to Tableau Server as part of the request for a view.
5. Tableau Server locates the string pair with the Base64 UUID, and then hashes the secret string to verify that it matches the hash stored in the repository.

This process ensures that any trusted ticket content stored on Tableau Server cannot be used to impersonate users or access content protected by authentication. However, because the full trusted ticket is sent over HTTP between Tableau Server and the client, the process relies on secure and encrypted transmission of HTTP data. Therefore, we recommend that you only deploy trusted tickets over SSL/TLS or another layer of network encryption.

Add Trusted IP Addresses or Host Names to Tableau Server

The first step in setting up trusted authentication is to configure Tableau Server to recognize and trust requests from one or more web servers:

Use the TSM web interface

1. Open TSM in a browser:

   https://<tsm-computer-name>:8850. For more information, see Sign in to Tableau Services Manager Web UI.

2. Click User Identity & Access on the Configuration tab and then click Trusted Authentication.

3. Under Trusted Authentication, for each trusted host, enter the name or IP address and then click Add:
Notes:
The values you specify completely overwrite any previous setting. Therefore, you must include the full list of hosts if you want to amend an existing list.

Static IP addresses are required: The web servers you specify must use static IP addresses, even if you use host names.

If you have one or more proxy servers between the computer that is requesting the trusted ticket (one of those configured in Step 2 as shown at Trusted Authentication) and Tableau Server, you also need to add them as trusted gateways using the `tsm configuration set gateway.trusted` option. See Configuring Proxies for Tableau Server for steps.

4. Enter a value in **Token Length** (Optional).

   The token length determines the number of characters in each trusted ticket. The default setting of 24 characters provides 144 bits of randomness. The value can be set to any integer between 9 and 255, inclusive.

5. Click **Save Pending Changes** after you've entered your configuration information.

6. Click **Pending Changes** at the top of the page:
7. Click **Apply Changes and Restart**.

Use the TSM CLI

1. **Enter the following command:**

   

   ```
   tsm authentication trusted configure -th <trusted IP address or host name>
   
   In the command above, `<trusted IP address>` should be a comma-separated list of the IPv4 addresses or host names of your web server(s), with each host name or IP address in quotes.
   
   **Note:** The values you specify completely overwrite any previous setting. Therefore, you must include the full list of hosts in the `tsm authentication trusted configure -th` command. (You cannot amend the list of hosts by running the `tsm authentication trusted configure -th` command repeatedly.)
   
   For example:
   
   ```
   tsm authentication trusted configure -th "192.168.1.101", "192.168.1.102", "192.168.1.103"
   
   or
   
   tsm authentication trusted configure -th "webserv1", "webserv2", "webserv3"
   ```
   
   **Notes:**
   
   Each host name or IP address in the list must be in double-quotes, followed by a
comma and one space after each comma.
The web servers you specify must use static IP addresses, even if you use host names.

2. If you have one or more proxy servers between the computer that is requesting the trusted ticket (one of those configured in Step 2 as shown at Trusted Authentication) and Tableau Server, you also need to add them as trusted gateways using the `tsm configuration set gateway.trusted` option. See Configuring Proxies for Tableau Server for steps.

3. Type the following command to save the changes to all the server configuration files:

   `tsm pending-changes apply`

   The `pending-changes apply` command displays a prompt to let you know this will restart Tableau Server if the server is running. The prompt displays even if the server is stopped, but in that case there is no restart. You can suppress the prompt using the `--ignore-prompt` option, but this does not change the restart behavior. For more information, see `tsm pending-changes apply`.

   There are other optional trusted authentication configurations (legacy support, logging, and timeout settings) that you can make by passing a json file to Tableau Server. See `trustedAuthenticationSettings Entity`.

Next, you need to configure your web server to receive tickets from Tableau Server.

Get a Ticket from Tableau Server

After you’ve added trusted IP addresses to Tableau Server, you’re ready to configure your web server to get tickets from Tableau Server via POST requests (step 3 in the diagram). The POST request must be sent to `http://<server name>/trusted`, not `http://tabserv`. For example `http://tabserv/trusted`. 
Here's the data you can use in a POST request to Tableau Server:

- **username=<username>** (required): The username for a licensed Tableau Server user. If you are using Local Authentication the username can be a simple string (for example, username=jsmith). If you are using Active Directory with multiple domains you must include the domain name with the username (for example, username=MyCo\jsmith).

- **target_site=<site_id>** (required if view not on Default site): Specifies the site containing the view if Tableau Server is running multiple sites and the view is on a site other than the Default site (for example, target_site=Sales). The value you use for <site_id> should be the Site ID that was provided when the site was created. This value is case sensitive. If the Site ID is SAles, then the target_site=e=SAles.

- **client_ip=<IP address>** (optional): Used to specify the IP address of the computer whose web browser is accessing the view (for example, client_ip=123.45.67.891). It is not the IP address of the web server making the POST request of Tableau Server. If you decide to use this parameter, see Optional: Configure Client IP Matching for more information.

Tableau Server’s response to the POST request will be a unique string (the ticket). If Tableau Server isn’t able to process the request, the return will be -1. See Ticket Value of -1 Returned from Tableau Server for tips on how to correct this. Also, in order for users to successfully authenticate when they click an embedded view, their browsers must be configured to allow third-party cookies.

The ticket format changed in Tableau Server 10.2. The ticket format is now a string composed of two parts. Each part is a 128 bit string that is encoded before it is returned to the client. The first part is a universally unique ID (UUID v4) that is Base64-encoded. The second
part is a 24-character secure random string. The concatenation of these parts can be expressed as Base64(UUIDv4):SecureRandomString. An example of a ticket might look like this: 9D1ObyqDQmSIyQpKdy4Sw==:dg62gCsSE0QRARXNTOp6mlJ5.

Next, you need to add code that allows the web server to construct an URL for the view that includes the view’s location and the ticket.

Display the View with the Ticket

After you create the POST request, you need to write code that provides the web server with the view’s location and the ticket from Tableau Server. It will use this information to display the view. How you specify it depends on whether the view is embedded, and if Tableau Server is running multiple sites.

Tableau Server View Examples

Here’s an example of how to specify a view that users only access via Tableau Server (the view is not embedded):

http://tabserver/trusted/<ticket>/views/<workbook>/<view>

If Tableau Server is running multiple sites and the view is on a site other than the Default site, you need to add t/<site ID> to the path. For example:

http://t-abservner/trusted/<ticket>/t/Sales/views/<workbook>/<view>

Use the same capitalization that you see in the Tableau Server URL.

Embedded View Examples

Here are some examples of how to specify embedded views. Because there are two approaches you can take with embed code, both ways are provided below. Regardless of which you use, there is some information unique to trusted authentication that you must provide. For more information, search for "Writing Embed Code" in the Tableau Server Help.
Note: The examples below use embed code parameters. For more information, see Embed Code Parameters in the Tableau Help.

**Script Tag Examples**

This example uses the **ticket** object parameter:

```html
<script type="text/javascript" src="http://myserver/javascripts/api/viz_v1.js"></script>
<object class="tableauViz" width="800" height="600" style="display:none;">
  <param name="name" value="MyCoSales/SalesScoreCard" />
  <param name="ticket" value="9D1ObyqDQmSIOyQpKdy4Sw==:dg62gCsSE0QRArXNTOp6mlJ5" />
</object>
```

Here’s what the above example looks like for a multi-site Tableau Server, where the view is published on the **Sales** site:

```html
<script type="text/javascript" src="http://myserver/javascripts/api/viz_v1.js"></script>
<object class="tableauViz" width="800" height="600" style="display:none;">
  <param name="site_root" value="/t/Sales" />
  <param name="name" value="MyCoSales/SalesScoreCard" />
  <param name="ticket" value="9D1ObyqDQmSIOyQpKdy4Sw==:dg62gCsSE0QRArXNTOp6mlJ5" />
</object>
```

Instead of using **ticket**, you can use the **path** parameter to state the full path of the view explicitly. When **path** is used, you do not also need the **name** parameter, which is usually a required parameter in Tableau JavaScript embed code:
Here's the same example, but for a multi-site server. Note that /t/<site ID> is used here:

```xml
<object class="tableauViz" width="900" height="700" style="display:none;">
  <param name="path" value="trusted/9D1ObyqDQmSI0yQpKdy4Sw==:dg62gCsSE0QRArXNTOp6mlJ5/t/Sales/views/MyCoSales/SalesScoreCard" />
</object>
```

**Iframe Tag Example**

```html
<iframe src="http://tabserver/trusted/9D1ObyqDQmSI0yQpKdy4Sw==:dg62gCsSE0QRArXNTOp6mlJ5/views/workbookQ4/SalesQ4?:embed=yes" width="800" height="600"></iframe>
```

**Optional: Configure Client IP Matching**

By default, Tableau Server does not consider the client web browser IP address when it creates or redeems tickets. To change this, you need to do two things: specify an IP address using the `client_ip` parameter in the POST request that obtains the ticket, and follow the steps below to configure Tableau Server to enforce client IP address matching.
1. Open a bash shell as an administrator and type the following command:

   tsm configuration set -k wgserver.extended_trusted_ip_checking -v true

2. Then type the following command:

   tsm pending-changes apply

   The `pending-changes apply` command displays a prompt to let you know this will restart Tableau Server if the server is running. The prompt displays even if the server is stopped, but in that case there is no restart. You can suppress the prompt using the `--ignore-prompt` option, but this does not change the restart behavior. For more information, see `tsm pending-changes apply`.

Test Trusted Authentication

The steps below provide a method you can use to test retrieving a trusted ticket from your web server. This simple test can help evaluate connectivity between the web server and Tableau Server, and whether or not trusted authentication has been configured correctly.

Alternatively, run a trusted ticket generator to test your configuration. The following URL references a trusted ticket generator that is not supported by Tableau. However the generator has been used by many customers to test their trusted ticket configuration: [https://github.com/mkannan-tsi/Trusted-Ticket-Generator](https://github.com/mkannan-tsi/Trusted-Ticket-Generator).

Step 1: Add a test user

Create a user on the Tableau Server that you can use to test trusted ticket functionality. See Add Users to Tableau Server. Add that user to a site on the server, and set the user's site role to **Explorer**.

Step 2: Create a test HTML page

Paste the following code into a new .html file that you save on the web server you specified in Step 1. You can change the labels and style attributes as you prefer.

```html
<html>
```
<head>
<title>Trusted Ticket Requester</title>
<script type="text/javascript">
    function submitForm()
    {
        document.getElementById('form1').action =
        document.getElementById('server').value + "/trusted";
    }
</script>
<style type="text/css">
    .style1 {width: 100%;}
    .style2 {width: 429px;}
    #server {width: 254px;}
</style>
</head>
<body>
<h3>Trusted Ticketer</h3>
<form method="POST" id="form1" onSubmit="submitForm()">
    <table class="style1">
        <tr>
            <td class="style2">Username</td>
            <td><input type="text" name="username" value="" /></td>
        </tr>
        <tr>
            <td class="style2">Server</td>
            <td><input type="text" id="server" name="server" value="https://" /></td>
        </tr>
        <tr>
            <td class="style2">Client IP (optional)</td>
            <td><input type="text" id="client_ip" name="client_ip" value="" /></td>
        </tr>
    </table>
</form>
</body>
Step 3: Retrieve a trusted ticket from Tableau Server

The following procedure will return a trusted ticket from Tableau Server.

1. Open the web page that you created in the previous step.

![Trusted Ticketer](image)
Tableau Server on Linux Administrator Guide

This operation requires JavaScript, so the web browser might prompt you to allow scripts to run.

2. In the text boxes, enter the following:
   - **Username**: The test user that was created in Step 2.
   - **Server**: the address of your Tableau Server, e.g., https://server_name.
   - **Client IP (optional)**: The IP address of the user’s computer, if it’s configured for client trusted IP matching.
   - **Site**: The name of the Tableau Server site that the test user is a member of.

3. Click **Get Ticket**. One of the following will be returned:
   - **A unique ticket**: A trusted ticket is a string composed of a base64-encoded UUID and a 24-character random string, for example, 9D10lxmDQmSIOyQpKdy4Sw==:dg62gCsSE0QRArXNTOp6mlJ5.
   - **-1**: If the value, -1 is returned, the configuration contains an error. See Ticket Value of -1 Returned from Tableau Server.

Step 4: Test access with trusted ticket

Now that you have a ticket, you can use it to access content on Tableau Server.

Construct a URL with the unique ticket that you generated in the previous step to verify access with the trusted ticket. The URL syntax is different if you are accessing a Tableau Server with a single site vs a server that hosts multiple sites.

**Single (Default) site server url**

https://<server-name>/trusted/<unique-ticket>/views/<workbook-name>/<view-name>

**Non-default site server url**

https://<server-name>/trusted/<unique-ticket>/t/<site-name>/views/<workbook-name>/<view-name>

Variables in the URLs are indicated by angle brackets (< and >). All other syntax is literal.
Troubleshoot Trusted Authentication

This section includes some common issues and errors you might encounter when you're configuring trusted authentication.

Trusted authentication information is written to /var/opt/tableau/tableau_server/data/tabsvc/logs/vizqlserver/vizql-*/*.log.

To increase the logging level, set the vizqlserver.trustedticket.log_level key to debug. For more information, see Change Logging Levels.

To test your trusted authentication deployment, see Test Trusted Authentication.

Ticket Value of -1 Returned from Tableau Server

Tableau Server returns -1 for the ticket value if it cannot issue the ticket as part of the trusted authentication process. Before troubleshooting this scenario, be sure to set the log level for trusted authentication to debug as specified in Troubleshoot Trusted Authentication.

The exact reason for this message is written to the vizqlserver_node*-*.log.* files in the following folder:

/var/opt/tableau/tableau_server/data/tabsvc/logs/vizqlserver

Here are some things to confirm:

- **All web server host names or IP addresses are added to trusted hosts**

  The log error, "Invalid request host: <IP address>" may indicate that the IP address or host name for the computer sending the POST request is not in the list of trusted hosts on Tableau Server. See Add Trusted IP Addresses or Host Names to Tableau Server to learn how to add IP addresses or host names to this list.

- **IP addresses are IPv4**

  If you are using IP addresses to specify trusted hosts, they must be in Internet Protocol version 4 (IPv4) format. An IPv4 address looks like this: 123.456.7.890. IPv6
addresses (for example, fe12::3c4a:5eab:6789:01c%34) are not supported as a way of inputting trusted hosts.

- **Username in POST request is a valid Tableau Server user**

  The username you send in the POST request must be a licensed Tableau Server user. You can see a list of users by signing in to Tableau Server as an administrator.

  The following log errors indicate a user POST issue:

  - "Missing username and/or client_ip"
  - "Invalid user: <username>"
  - "Unlicensed user is not allowed: <username>"

- **Username in POST request includes domain**

  If Tableau Server is configured to use Local Authentication, the username that you send in the POST can be a simple string. However, if the server is configured for Active Directory you must include the domain name with the user name (domain\username). For example, the username parameter might be: `username=dev\jsmith`. A common error log for this scenario is "Invalid user: <username>.

- **Content-Type is specified**

  If you are designing an ASP.NET or C# application, you need to declare the content type in your HTTP request. For example:

  ```
  http.setRequestHeader("Content-Type","application/x-www-form-urlencoded;charset=UTF-8")
  ```

  If you do not specify content type and Tableau Server returns a -1, the log files contain the error: "missing username and/or client_ip".
HTTP 401 - Not Authorized

If you receive a 401- Not Authorized error, you may have configured Tableau Server to use Active Directory with SSPI. If your web server uses SSPI, you do not need to set up trusted authentication. You can embed views and your users will have access to them as long as they are licensed Tableau server users and members of your Active Directory.

See tsm authentication sspi <commands>.

If you see a 401 error (or a 302 - Redirect error) after you have deployed Tableau Server 2019.4, then it’s likely the trusted ticket code you have written to construct the URL for the client has not been updated to account for the two-part ticket URL format.

See Get a Ticket from Tableau Server.

HTTP 404 - File Not Found

You may receive this error if your program code references a Tableau Server URL that does not exist. For example, your web server may construct an invalid URL that cannot be found when the webpage tries to retrieve it.

Another cause for this error is if you did not enter the trusted web servers as specified in Add Trusted IP Addresses or Host Names to Tableau Server. If you have already entered the trusted web servers, verify that the IP addresses or host names are correct.

Invalid User (SharePoint or C#)

You may encounter this error if you’ve configured Tableau Server for trusted authentication.

The example code for the SharePoint .dll references the following GET request:


The above request will return the display name of the current Windows Active Directory user. If you want to use the login ID, then you will need to change the code to:


After you make the change, recompile the SharePoint .dll.
Attempting to Retrieve the Ticket from the Wrong IP Address

You may encounter this error if you've configured Tableau Server for trusted authentication.

The client web browser IP address is not considered by default when redeeming the ticket. If Tableau Server is configured to enforce client IP address matching, make sure that the client’s web browser IP address that is sent in the POST to Tableau Server is the same as when the browser tries to retrieve the embedded view. For example, in the Trusted Authentication diagram, if the POST request in step 3 sends the parameter client_ip=74.125.19.147, then the GET request in step 5 must come from that same IP address.

See Optional: Configure Client IP Matching to learn how to configure Tableau Server to enforce client IP address matching.

Cookie Restriction Error

When a user signs in to Tableau Server, a session cookie is stored in their local browser. The stored cookie is how Tableau Server maintains that the signed in user has been authenticated and can access the server. Because the cookie is set with the same domain or subdomain as the browser’s address bar, it is considered a first-party cookie. If a user’s browser is configured to block first-party cookies, they will be unable to sign in to Tableau Server.

When a user signs in to Tableau Server via an embedded view, or in an environment where trusted authentication has been configured, the same thing happens: a cookie is stored. In this case, however, the browser treats the cookie as a third-party cookie. This is because the cookie is set with a domain that’s different from the one shown in the browser’s address bar. If a user’s web browser is set to block third-party cookies, authentication to Tableau Server will fail. To prevent this from occurring, web browsers must be configured to allow third-party cookies.

An error occurred communicating with the server (403)

If Tableau Server is configured for trusted authentication, you may receive this error after opening a new view in a browser and attempting to navigate back to views you’d opened earlier. Tableau Server provides protection against unauthorized reuse of VizQL sessions through the tsm configuration set Optionsvizqlserver.protect_sessions, which is
set to `true` by default. Because Tableau Server is configured for trusted authentication, you may not also need to enable `vizqlserver.protect_sessions`. To disable it, use `tsm configuration set Options` to change it to `false`.

**Personal Access Tokens**

Personal access tokens provide Tableau Server users the ability to create long-lived authentication tokens. The tokens allow users to run automation with Tableau REST APIs without requiring hard-coded credentials or interactive login. More information about using personal access tokens with Tableau REST APIs is at [Signing In and Out (Authentication)](https://example.com).

Personal access tokens are not used for generic client access to the Tableau Server web interface, TSM, or tabcmd interfaces.

We recommend creating personal access tokens for automated scripts and tasks that are created with Tableau REST API:

- **Improving security**: Personal access tokens reduce risk in the event credentials are compromised. In the case where Tableau Server uses Active Directory or LDAP as an identity store, you can reduce the scope of credential compromise by using a personal access token for automated tasks. In this case, using an application-specific token doesn't expose the broader system in the event that automation or script files are compromised. If a token gets compromised or is used in automation that is failing or posing a risk, you can just revoke the token. You do not need to rotate or revoke the user's credentials.

- **Auditing and tracking**: As an administrator, you can review Tableau Server logs to track when a token is used, what sessions are created from that token, and the actions that are performed in those sessions. You can also determine if a session and the related tasks were performed from a session that was generated from a token or from an interactive login.

- **Managing automation**: A token can be created for each script or task that is run. This allows you to silo and review automation tasks across your organization. Additionally, by using tokens then password resets or metadata changes (username, email, etc.) on user accounts will not disrupt automation as it would when credentials are hard-coded into the scripts.
Understanding personal access tokens

When a token is created, it is hashed then stored in the repository. After the token is hashed and stored, the original token is deleted. Users are instructed to copy the token to a safe place and to handle it as they would a password. When the token is used at run-time, Tableau Server hashes the token presented by the user and compares it to the hashed value stored in the repository. If a match is made, then an authenticated session is started.

In the context of authorization, Tableau Server handles the authenticated session with same permissions and rights that the user has as an interactive user.

Creating tokens

Users with accounts on Tableau Server can create, manage, and revoke personal access tokens on the My Account Settings page. See Manage Your Account Settings in the Tableau Desktop and Web Authoring Help for more information.

Users must create their own personal access tokens. Administrators cannot create tokens for users.

Revoking users' tokens

Users are able to revoke their own tokens on the My Account Settings page. As an administrator, you can also revoke personal access tokens.

1. Sign in to the Tableau Server Admin Area as a site administrator or server administrator.
2. Locate the user whose token you want to revoke. For more information about navigating Server Admin pages and locating users, see View, Manage, or Remove Users.
3. Click the user’s name to open their profile page.
4. On the user’s profile page, click the Settings tab.
5. In the Personal Access Tokens section, identify the token that you want to revoke and then click Revoke.
6. On the verification pop-up, click Delete.
Tracking and monitoring usage

All token-related actions are logged in the Tableau Server Application Server (vizportal) service.

To locate token-related activities, filter log entries containing the string, RefreshTokenService.

Tokens are stored in this format: Token Guid: <TokenID(Guid)>, where the TokenID is a base64 encoded string. The token secret is not included in the logs. For example:

Token Guid: 49P+CxmARY6A2GHxyvHHAA== (e3d3fe0b-1980-458e-80d8-61f1caflc700).

The following is an example snippet of two log entries. The first shows how a user is mapped to a token. The second shows a refresh event for the same token:

RefreshTokenService - Issued refresh token to the following user: jsmith. Token Guid: 49P+CxmARY6A2GHxyvHHAA== (e3d3fe0b-1980-458e-80d8-61f1caflc700)
RefreshTokenService - Redeemed refresh token. Token Guid: 49P+CxmARY6A2GHxyvHHAA== (e3d3fe0b-1980-458e-80d8-61f1caflc700)

To locate key operations, filter log entries containing the string, OAuthController.

Data Connection Authentication

You can configure data connection authentication using Kerberos, OAuth, and single sign-on.

More information

- Tableau Server on Linux - Connecting to a Windows Shared Directory (Tableau Community)
Setting an Oracle Connection to Use TNSNames.ora or LDAP.ora (Tableau Support)

Enable Kerberos Delegation

Kerberos delegation enables Tableau Server to use the Kerberos credentials of the viewer of a workbook or view to execute a query on behalf of the viewer. This is useful in the following situations:

- You need to know who is accessing the data (the viewer’s name will appear in the access logs for the data source).
- Your data source has row-level security, where different users have access to different rows.

Supported data sources

Tableau supports Kerberos delegation with the following data sources:

- Cloudera: Hive/Impala
- Denodo
- Hortonworks
- PostgreSQL
- Spark
- SQL Server
- Teradata

MSAS is not supported on Linux platforms. Oracle delegation is not supported with Tableau Server on Linux because of driver limitations.

Requirements

Kerberos delegation requires Active Directory.

- The Tableau Server information store must be configured to use LDAP - Active Directory.
- The computer where Tableau Server is installed must be joined to Active Directory domain.
- MIT KDC is not supported.
Configuration process

This section provides an example of the process to enable Kerberos delegation. The scenario also includes example names to help describe the relationships between the configuration elements.

1. Tableau Server will need a Kerberos service ticket to delegate on behalf of the user that is initiating the call to the database. You must create a domain account that will be used to delegate to the given database. This account is referred to as the Run As service account. In this topic, the example user configured as the delegation/Run As account is tabsrv@example.com.

   The account must be configured with Active Directory User and Computers on a Windows Server that is connected to the user domain:

   - Open the Properties page for the Run As service account, click the Delegation tab and select Trust this user for delegation to specified services only and Use any authentication protocol.

2. Create a keytab file for the Run As service account.

   For example, the following commands create a keytab (tabsrv-runas.keytab) using the ktutil tool:

   ```
   sudo ktutil
   ktutil: addent -password -p tabsrv@EXAMPLE.COM -k 2 -e <encryption scheme>
   ```

   Encryption schemes for this command include RC4-HMAC, aes128-cts-hmac-shal-96, and aes256-cts-hmac-shal-96. Consult your IT team for the correct encryption scheme for your environment and data source.

   ```
   ktutil: wkt tabsrv-runas.keytab
   ```

   Tableau Server will use the Run As service account and the associated keytab to authenticate and make a direct connection to the database.
3. Copy the keytab into the Tableau Server data directory and set proper ownership and permissions.

```bash
mkdir /var/opt/keytab
sudo cp -p tabsrv-runas.keytab /var/opt/keytab

sudo chown $USER /var/opt/keytab/tabsrv-runas.keytab
chgrp tableau /var/opt/keytab/tabsrv-runas.keytab
chmod g+r /var/opt/keytab/tabsrv-runas.keytab
```

4. Run the following TSM commands to enable Kerberos delegation, set the delegation service account, and associate the keytab file with the service account:

```bash
tsm configuration set -k wgserver.delegation.enabled -v true
tsm configuration set -k native_api.datasource_impersonation_runas_principal -v tabsrv@example.com
tsm configuration set -k native_api.datasource_impersonation_runas_keytab_path -v <path-to-file>kerberos.keytab
tsm configuration set -k native_api.protocol_transition_a_d_short_domain -v false
tsm configuration set -k native_api.protocol_transition_uppercase_realm -v true
```

In some cases, TSM may return an error mentioning `--force-keys`. If you get this error, run the command again with the `--force-keys` parameter appended to the argument.

5. Run the following TSM command apply the changes to Tableau Server:

```
tsm pending-changes apply
```
The `pending-changes apply` command displays a prompt to let you know this will restart Tableau Server if the server is running. The prompt displays even if the server is stopped, but in that case there is no restart. You can suppress the prompt using the `--ignore-prompt` option, but this does not change the restart behavior. For more information, see `tsm pending-changes apply`.

6. Enable delegation for data connections:

- **Cloudera**—See [Enable Kerberos Delegation for Hive/Impala](#) in the Tableau Community.

- **Denodo**—See [Enabling Kerberos Delegation for Denodo on Linux](#) in the Tableau Community.

- **PostgreSQL**—See [Enabling Kerberos Delegation for PostgreSQL](#) in the Tableau Community.

- **SQL Server**—See [Enabling Kerberos Delegation for SQL Server](#) in the Tableau Community.

- **Teradata**—See [Enabling Kerberos Delegation for Teradata](#) in the Tableau Community.

See also

Troubleshoot Kerberos OAuth Connections

For Google BigQuery, Google Analytics, Salesforce, OneDrive, Dropbox, and QuickBooks Online, an alternative to storing your sensitive database credentials with Tableau Server is to create connections using the **OAuth 2.0** standard.

From Tableau, when you sign in to data with a provider that uses OAuth, you are redirected to the provider’s sign-in page. After you provide your credentials and authorize Tableau to access your data, the data provider sends Tableau an **access token** that uniquely identifies requests from Tableau. For more information, see Overview of the OAuth process below.
Using OAuth connections provides the following benefits:

- **Security**: Your database credentials are never known to or stored in Tableau Server, and the access token can be used only by Tableau.

- **Convenience**: Instead of having to embed your data source ID and password in multiple places, you can use the token provided for a particular data provider for all published workbooks and data sources that access that data provider.

  In addition, for live connections to Google BigQuery data, each workbook viewer can have a unique access token that identifies the user, rather than sharing a single user name and password credential.

Overview of the OAuth process

The following steps describe a workflow in the Tableau environment that calls the OAuth process.

1. You take an action that requires access to a cloud data source.
   
   For example, you open a workbook that’s published to Tableau Server.

2. Tableau directs you to the cloud data provider’s sign-in page. The information that is sent to the data provider identifies Tableau as the requesting site.

3. When you sign in to the data, the provider prompts you to confirm your authorization for Tableau to access the data.

4. Upon your confirmation, the data provider sends an access token back to Tableau Server.
5. Tableau Server presents your workbook and data to you.

The following workflows can use the OAuth process:

- Creating a workbook and connecting to the data source from Tableau Desktop or from Tableau Server.

- Publishing a data source from Tableau Desktop.

- Creating a flow and connecting to the data source from Tableau Prep Builder version 2019.3.1 or later.

- Signing in to Tableau Server from an approved client, such as Tableau Mobile or Tableau Desktop.

Access tokens for data connections

You can embed credentials based on access tokens with data connections, to enable direct access after the initial authentication process. An access token is valid until a Tableau Server user deletes it, or the data provider revokes it.
It is possible to exceed the number of access tokens your data source provider allows. If that's the case, when a user creates a new token, the data provider uses length of time since last access to decide which token to invalidate to make room for the new one.

Access tokens for authentication from approved clients

By default, Tableau Server sites allow users to access their sites directly from approved Tableau clients, after users provide their credentials the first time they sign in. This type of authentication also uses OAuth access tokens to store the users' credentials securely.

For more information, see Disable Automatic Client Authentication

Configure the Server for OAuth Support

Instead of individual usernames and passwords, OAuth works through limited-purpose access tokens. Before you can obtain access tokens needed to create an OAuth connection in Tableau, you need to configure your server so that the data provider sending the token can recognize Tableau Server as a trusted destination. The following section describes how to prepare for setting up OAuth regardless of your data provider. The topics listed below it contain the steps for configuring your server for specific data providers.

Preparing for Configuring OAuth Support

Before you begin the configuration steps specific to your data provider, complete the following prerequisites:

- Obtain the fully qualified domain name of each Tableau Server node that will host views that connect to this data source. For example:

  https://sales.your_domain.com

  If you use Salesforce.com, you will need to provide an https address.

- Make sure at least one of your data-provider accounts is enabled for API access.
For **Google** data types, you need access to the developers console on the Google Cloud Platform.

For **Salesforce.com**, you need access to the Force.com platform.

For **QuickBooks Online**, you need access to the Intuit platform.

## Configure Settings for Your Data Provider

When you complete the OAuth-preparation steps, you can configure the appropriate settings with your data provider.

- Set up OAuth for Google
- Set up OAuth for Salesforce.com
- Set up OAuth for QuickBooks Online

### Set up OAuth for Google

This topic describes how to set up your Google BigQuery, Google Sheets, and Google Analytics data sources for OAuth.

Complete these steps for each Tableau Server instance.

**Note** Before you complete these steps, make sure you have completed the prerequisites described in Preparing for Configuring OAuth Support.

Set up OAuth by following these three procedures:

- Enable API access and create access token from Google.
- Configure OAuth on Tableau Server.
- Create and edit Google data source.
Obtain a Client ID and Enable Google APIs

Note These steps reflect the settings in the Google Cloud Platform console at the time of this writing. For more information, see Using OAuth 2.0 for Web Server Applications in the Google Developers Console Help.

1. Sign in to Google Cloud Platform, and then click Go to my console.

2. On the drop-down menu next to the Google Cloud Platform title, select Create project.

3. In the new project form that appears, complete the following:
   - Give the project a meaningful name that reflects the Tableau Server instance for which you’ll use this project.
   - Determine whether you want to change the project ID.

   Note After you create the project, you will not be able to change the project ID. For information, click the question mark icons.

4. Open the new project, and navigate to APIs Manager > Credentials.
Note: You must generate credentials with the Google API Manager for Tableau Server. Do not attempt to use a Service Account, which generates a key, and is not supported.

5. Click the OAuth consent screen tab and then enter a meaningful name for Product name shown to users.

6. Click the Credentials tab, then click the Create credentials drop-down list, and then select OAuth client ID. Complete the following:
   - Select Web Application.
   - For Authorized JavaScript Origins, enter the local computer name of your Tableau Server.
   - For Authorized Redirect URI, replace the existing text with the Internet address for your server, and add the following text to the end of it: auth/add_oauth_token. For example:

     https://your_server_url.com/auth/add_oauth_token

7. Copy the Authorized Redirect URI, and paste it in a location that you can access from your Tableau Server computer.

8. Click Create.

9. Copy the following values that Google returns, and paste them in a location that you can access from your Tableau Server computer:
   - Client ID
   - Client secret

10. In APIs Manager > Dashboards, verify that BigQuery API, Google Drive API (to
enable Google Sheets) or Analytics API is enabled. To enable APIs, click ENABLE API at the top of the page.

**Configure Tableau Server for Google OAuth**

Using the information you obtained by completing the steps in Obtain a Client ID and Enable Google APIs, configure your Tableau Server:

- On the Tableau Server computer, open the shell and run the following commands to specify the access token and URI:

  ```
  tsm configuration set -k oauth.google.client_id -v <your_client_ID>
  tsm configuration set -k oauth.google.client_secret -v <your_client_secret>
  tsm configuration set -k oauth.google.redirect_uri -v <your_authorized_redirect_URI>
  tsm pending-changes apply
  ```

  The `pending-changes apply` command displays a prompt to let you know this will restart Tableau Server if the server is running. The prompt displays even if the server is stopped, but in that case there is no restart. You can suppress the prompt using the `-ignore-prompt` option, but this does not change the restart behavior. For more information, see tsm pending-changes apply.

**Create and edit Google data source**

Next, you must publish the Google data sources to the server. See the Tableau Desktop topic, Google BigQuery.
After you’ve published the data sources, the final step is to edit the data source connection to use the embedded access token that you configured earlier. See Edit Connections on Tableau Server.

Managing access tokens

After you configure the server for OAuth, you can allow users to manage their own access tokens in their profile settings, or you can manage the tokens centrally. For more information, see Allow Saved Access Tokens.

Set up OAuth for Salesforce.com

This topic describes how to set up your Salesforce.com data sources for OAuth. Complete these steps for each Tableau Server instance.

**Note:** Before you complete these steps, make sure you have completed the prerequisites described in Preparing for Configuring OAuth Support.

Set up OAuth by following these two procedures:

- Create a Connected App in Salesforce
- Use the information you obtained to configure your server.

Create a Connected Salesforce App

1. Sign in to your Salesforce.com developer account, click your user name in the upper-right, and then select **Setup**.
2. In the left navigation column, under App Setup, select Create > Apps.

3. In the Connected Apps section, click New.

4. Complete the Basic Information, and in the API section, select Enable OAuth Settings.

5. In the new OAuth settings that appear, for Callback URL, type the fully qualified domain name of your server, using the https protocol, and append the following text to the URL: auth/add_oauth_token.

   For example:

   https://www.your_server.com/auth/add_oauth_token

6. Move the following items from Available OAuth Scopes to Selected OAuth Scopes:

   - Access and manage your data (api)
   - Access your basic information (id)
   - Perform requests on your behalf at any time (refresh_token)
7. Click **Save**.

After you save the app, Salesforce populates the API section with the following IDs that you will use to configure Tableau Server:

- Consumer Key
- Consumer Secret
- Callback URL

**Configure Tableau Server for Salesforce.com OAuth**

1. On the Tableau Server computer, run the following commands:

   ```
   tsm configuration set -k oauth.salesforce.client_id -v <your_client_ID>
   tsm configuration set -k oauth.salesforce.client_secret -v <your_client_secret>
   tsm configuration set -k oauth.salesforce.redirect_uri -v <your Authorized_redirect_URI>
   ```

2. (Optional) To change the default login server, type the following command:
3. Enter the following command to apply changes:

```
 tsm pending-changes apply
```

The `pending-changes apply` command displays a prompt to let you know this will restart Tableau Server if the server is running. The prompt displays even if the server is stopped, but in that case there is no restart. You can suppress the prompt using the `--ignore-prompt` option, but this does not change the restart behavior. For more information, see `tsm pending-changes apply`.

### Managing access tokens

After you configure the server for OAuth, you can allow users to manage their own access tokens in their profile settings, or you can manage the tokens centrally. For more information, see Allow Saved Access Tokens.

### Set up OAuth for QuickBooks Online

This topic describes how to set up your QuickBooks Online data sources for OAuth authentication. Complete these steps for each Tableau Server instance.

**Note:** Before you complete these steps, make sure you have completed the prerequisites described in Preparing for Configuring OAuth Support.

Setting up OAuth for QuickBooks Online consists of two tasks:

- Create a Connected App on the Intuit developer platform.

- Use the information you get as part of the Connected App to configure your server.
Create a Connected Intuit App

1. Sign in to your Intuit developer account, and then click My Apps.

2. In the Just start coding section, click Select APIs.

3. Select Accounting and click Create App.

4. In the Get your app ready for submission section, click the link to get your production keys.

   **Important:** You must use production keys rather than development keys.

5. Copy the app token, OAuth consumer key, and OAuth consumer secret.

Configure Tableau Server for QuickBooks Online

- On the Tableau Server computer, open the bash shell and run the following commands:

  ```
  tsm configuration set -k oauth.quickbooks.oauth_callback_uri -v http://YOUR-SERVER/auth/add_oauth_token
  
  tsm configuration set -k oauth.quickbooks.consumer_key -v <your_consumer_key>
  
  tsm configuration set -k oauth.quickbooks.consumer_secret -v <your_consumer_secret>
  
  tsm pending-changes apply
  
  The pending-changes apply command displays a prompt to let you know this will restart Tableau Server if the server is running. The prompt displays even if the
server is stopped, but in that case there is no restart. You can suppress the prompt using the \texttt{--ignore-prompt} option, but this does not change the restart behavior. For more information, see tsm pending-changes apply.

Managing access tokens

If you run an extract refresh job for your QuickBooks Online data source, Tableau Server attempts to renew access tokens for you. To help ensure that your access tokens do not expire, run your extract refresh jobs more than once a month. Otherwise, the access tokens from QuickBooks Online expire and your extract refresh jobs fail. If your access tokens do expire, you can edit your saved credentials from the \texttt{Settings} page.

The saved credentials can be managed centrally or by your users. For more information, see Allow Saved Access Tokens.

Allow Saved Access Tokens

After you configure Tableau Server for OAuth, you can decide to allow users to manage their own OAuth credentials, or you want to manage them centrally. If you want users to manage their own, you need to enable user profile settings from the server.

\textbf{Note:} If you have not yet configured your server to enable OAuth data connections, see the related topics listed below.

1. Sign in to Tableau Server as a server administrator.

2. \textbf{Single-site:} Click \texttt{Settings} > \texttt{General}.

   \textbf{Multisite:} In the site menu, click \texttt{Manage All Sites} and then click \texttt{Settings} > \texttt{General}.

3. In the \texttt{Saved Credentials} section, select the following:
- Allow users to save passwords for data sources (allows users to save their individual credentials with data sources).

- Allow users to save OAuth access tokens for data sources

![Saved Credentials](image)

4. Click **Save**.

After you select these check boxes, users will see a **Manage Credentials** section in their profile settings, where they can add access tokens for OAuth data connections.

![Manage Credentials](image)

### Managing credentials centrally

Server administrators alternatively can manage OAuth credentials centrally. This can work well, for example, if multiple users work from the same data, and you have a dedicated user account for your data provider.

To manage credentials centrally, you do the following:
Tableau Server on Linux Administrator Guide

- Clear the check boxes described in the preceding procedure.

- Edit connection information as data sources are published.

  When you edit the connection, you embed credentials that use an OAuth access token instead of an individual’s user name and password.

When the settings for saving passwords and access tokens are not enabled, the Manage Credentials section is excluded from users’ profile settings.

See also

Set up OAuth for Google

Set up OAuth for Salesforce.com

Set up OAuth for QuickBooks Online

Troubleshoot OAuth Connections

This topic provides information about resolving issues that can occur when you configure OAuth data connections.

Conflict error

In some cases, users may receive an error when attempting to connect with OAuth. The first sentence of the error message is:

*The server encountered an internal error or misconfiguration and was unable to complete your request.*

This error indicates that the fully qualified domain name (FQDN) of the Tableau Server needs to be added to the whitelist redirect key on Tableau Server.

When users are accessing a Tableau Server by the local host name (https://tableau) and the OAuth data provider is responding to the public DNS name (https://data.example.com),
Tableau Server must associate the external FQDN with the local server name. The local host name is the server name in the URL that users enter when accessing Tableau Server from the internal network.

To fix this error, run `tsm configuration set` with the `oauth.whitelisted.redirect_to_origin_host` key option. This key takes a value pair, "internal_host,FQDN1,FQDN2". For example, the following commands set the local host name to `tableau` and the FQDN to `tableau.example.com`:

```
tsm configuration set -k oauth.whitelisted.redirect_to_origin_host -v "tableau,tableau.example.com"

tsm pending-changes apply
```

In the case where multiple public URLs are used to access the internal Tableau Server, add additional FQDNs to the command, separated by commas, for example:

```
tsm configuration set -k oauth.whitelisted.redirect_to_origin_host -v "tableau,tableau.example.com,tableau2.example.com"
```

If you need to edit an existing whitelist redirect configuration, you must enter the full mapping set. You cannot truncate or append existing configuration keys.

Configure SAP HANA SSO

You can configure Tableau Server to use SAML delegation to provide a single sign-on (SSO) experience for SAP HANA. This scenario is not dependent on SAML authentication to Tableau Server. You do not need to use SAML sign on with Tableau Server in order to use HANA SAML delegation. You can sign in to Tableau Server using whatever method you choose.

With SAML delegation for SAP HANA, Tableau Server functions as an identity provider (IdP).
Before you begin

Configuring SAML delegation with SAP HANA requires configuration on both Tableau Server and on SAP HANA. This topic provides configuration information about configuring Tableau Server. Before you configure Tableau Server, you must complete the following:

- Acquire a SAML certificate and key file for Tableau Server. The certificate file must be a PEM-encoded x509 certificate with the file extension .crt or .cert. This file is used by Tableau Server, and must also be installed on HANA. The private key must be a DER-encoded private key file that is not password protected, and that has the file extension .der. This file is only used by Tableau Server.
- Install the certificate in HANA.
- Install the latest version of SAP HANA driver (minimum version is 1.00.9) on Tableau Server.
- Configure network encryption from Tableau Server to SAP HANA (recommended).

For more information about generating the certificate/key pair, encrypting the SAML connection, and configuring SAP HANA, see How to Configure SAP HANA for SAML SSO with Tableau Server in the Tableau Community.

Configure Tableau Server SAML for SAP HANA

The following procedure describes how to configure SAML for SAP HANA on Tableau Server using `tsm data-access`. You can also configure SAML for SAP HANA using the `sapHanaSettings` Entity.

If you are running Tableau Server in a distributed deployment, run the following procedure on the initial node.

1. Place certificate files in a folder named `saml`. For example:

   `/var/opt/saml`

2. Run the following commands to specify the location of the certificate and key files:

   `tsm data-access set-saml-delegation configure --cert-key <cert-key> --cert-file <cert-file>`
Where `<cert-key>` and `<cert-file>` are file paths to the private key and certificate file, respectively.

For example,

```
tsm data-access set-saml-delegation configure --cert-key /var/opt/saml/hana_pkey_pkcs8.der --cert-file /var/-opt/saml/hana_cert.pem
```

You can specify other options. For example, you can specify user name format and how credentials are normalized. See tsm data-access.

3. Run the following commands to enable delegation:

```
tsm data-access set-saml-delegation enable
```

```
tsm configuration set -k wgserver.sap_hana_sso.enabled -v true
```

```
tsm configuration set -k wgserver.delegation.enabled -v true
```

4. When you have finished, run `tsm pending-changes apply`.

   The `pending-changes apply` command displays a prompt to let you know this will restart Tableau Server if the server is running. The prompt displays even if the server is stopped, but in that case there is no restart. You can suppress the prompt using the `--ignore-prompt` option, but this does not change the restart behavior. For more information, see `tsm pending-changes apply`.

Enable Kerberos Service Account Access

You can configure Tableau Server to use a Kerberos service account to access a database. In this scenario, Tableau Server connects to databases with a service account, also referred to as a "RunAs account".
To use RunAs authentication on Tableau Server you must first create a workbook or data-source that uses integrated authentication. When users publish to Tableau Server they will get the option for RunAs authentication. If you create a datasource with Tableau Server web authoring that uses integrated authentication, the datasource will use RunAs authentication by default.

**Note:** *Integrated authentication* is also referred to as *Windows Authentication* on some connectors. In both cases, Tableau Server uses Kerberos authentication.

Data Access with the RunAs Service Account

To use RunAs authentication, the RunAs account requires read and query permissions to external databases. As designed, Tableau Server users with the *Creator* role or the *Explorer (Can Publish)* role have full access to the RunAs account for queries to external databases.

For example, a user with the Creator role can view all databases that have been granted access to the RunAs service account. They can also list tables and run Custom SQL.

If the Creator-user specifies the database host name and selects Integrated Authentication when creating a new data source with web authoring, then databases that have been granted RunAs access will be displayed to the user.

View access to database assets are not restricted to users who connect to Tableau Server with web authoring. Sophisticated users, who have the same roles noted above and who have knowledge of database server names, could create workbooks with Tableau Desktop that display databases that have been granted RunAs access.

**Recommendations**

Whether user access to databases in these scenarios is acceptable must be assessed by your organization. Generally, reducing the usage and scope of the RunAs service account will reduce the likelihood of inadvertent user access to database content. However, reducing
the usage and scope of the RunAs service account may also impose more credential management to you and your users.

Evaluate the following recommendations in context of your business needs and data access policies.

- Firstly, be sure that you trust all users who have Creator roles or Explorer (Can Publish) roles. You will rely on these users to perform actions in Tableau with integrity.
- If you cannot trust all of your users who have publishing rights on data sources that are accessed by the RunAs service account, then you should consider embedding credentials for those data sources.
- If a data source is not set up for automated extract refreshes, that is, the data source is primarily accessed as a live connection, then you may be able to use Kerberos Delegation. For requirements, see Enable Kerberos Delegation.

Requirements

- MIT Kerberos is not supported.
- The RunAs service account must have read access to the target database.

Configuration process

This section provides an example of the process to enable Kerberos service account access.

1. Create a domain user account to act as the RunAs service account. This account must have read access to the target database.

   In the example here, the RunAs service account is User principal named tab-srv@example.com.

2. Create a keytab file for the RunAs service account.

   For example, the following commands create a keytab \( \text{tabsrv-runas.keytab} \) using the ktutil tool:

   \( \text{ktutil} \)
Tableau Server on Linux Administrator Guide

ktutil: addent -p tabsrv@example.com -k 2 -e <encryption scheme>

Encryption schemes for this command include RC4-HMAC, aes128-cts-hmac-shal-96, and aes256-cts-hmac-shal-96. Consult your IT team for the correct encryption scheme for your environment and data source.

ktutil: wkt tabsrv-runas.keytab

Tableau Server will use the RunAs service account and the associated keytab to authenticate and make a direct connection to the database.

3. Copy the keytab into the Tableau Server data directory and set proper ownership and permissions. The keytab should be readable by the unprivileged user. The default unprivileged user created by Tableau Setup is tableau.

If you are running a multi-node deployment, then you must run the following commands on each node in the cluster:

mkdir /var/opt/tableau/tableau_server/keytab
sudo cp -p tabsrv-runas.keytab /var/opt/tableau/tableau_server/keytab
sudo chown $USER /var/opt/tableau/tableau_server/keytab/tabsrv-runas.keytab
chgrp tableau /var/opt/tableau/tableau_server/keytab/tabsrv-runas.keytab
chmod g+r /var/opt/tableau/tableau_server/keytab/tabsrv-runas.keytab

4. Run the following TSM commands to enable RunAs access, set the RunAs service account, and associate the keytab file with the service account.

tsm configuration set -k features.RunAsAuthLinux -v true --force-keys
tsm configuration set -k native_api.datasource_runas_
principal -v tabsrv@example.com --force-keys
tsm configuration set -k native_api.datasource_runas_keytab_path -v /var/opt/tableau/tableau_server-/keytab/tabsrv-runas.keytab --force-keys

5. Run the following TSM command apply the changes to Tableau Server deployment:

tsm pending-changes apply

The pending-changes apply command displays a prompt to let you know this will restart Tableau Server if the server is running. The prompt displays even if the server is stopped, but in that case there is no restart. You can suppress the prompt using the --ignore-prompt option, but this does not change the restart behavior. For more information, see tsm pending-changes apply.

SQL Server Impersonation

Impersonation in the context of Tableau Server means allowing one user account to act on behalf of another user account. You can configure Tableau and Microsoft SQL Server to perform database user impersonation, so that the SQL Server database account used by Tableau Server queries on behalf of SQL Server database users, who are also Tableau users.

The main benefit of this feature is it allows administrators to implement and control their data security policy in one place: their databases. When Tableau users access a view with a live connection to a SQL Server database, the view only displays what the users' database permissions authorize them to see. An additional benefit is that the users don't have to respond to a database sign-in prompt when they open the view. Also, workbook publishers don't have to rely on user-specific filters to restrict what's seen in views.

Impersonation Requirements

Here's what you need to use feature:
Live connections to SQL Server only: Impersonation can only be used for views that have a live connection to a SQL Server database, version 2005 or newer.

Individual database accounts: Each person who’ll be accessing the view must have an explicit, individual account in the SQL Server database to which the view connects. Members of an Active Directory (AD) group cannot be impersonated. For example, if Jane Smith is a member of the AD group Sales, and her database administrator adds the Sales AD group to the SQL Server database, Jane cannot be impersonated.

Matching credentials and authentication type: The credentials of each Tableau user's account and their Tableau user authentication type must match their credentials and authentication type in the SQL Server database. For example, if Jane Smith’s Tableau Server user account is MyCo\jsmith, the username on the SQL Server database must also be MyCo\jsmith. SQL Server must be using Windows Integrated Authentication.

SQL Server prerequisites: In SQL Server you should have a data security table, a view that enforces data security, and you should require that your database users use the view.

SQL IMPERSONATE account: You need a SQL Server database account that has IMPERSONATE permission for the above database users. This is either an account with the sysadmin role or one that has been granted IMPERSONATE permission for each individual user account (see the MSDN article on EXECUTE AS). The SQL Server account must be one of the following:

- The Tableau Server Run As service account. See Enable Kerberos Service Account Access.
- The workbook publisher's account. See Impersonate with Embedded SQL Credentials.

How Impersonation Works

Here’s an illustration of how database user impersonation works:
In the above illustration, Jane Smith (MyCo\jsmith) is a West Coast sales representative and Henry Wilson (MyCo\hwilson) covers the East. In the SQL Server database, the account permissions for Jane’s account, MyCo\jsmith, only give her access to West Coast data. Henry’s account, MyCo\hwilson, can only access data for the East Coast.

A view has been created that displays data for the entire country. It has a live connection to a SQL Server database. Both users sign in to Tableau Server and click the view. Tableau Server connects to SQL Server using a database account with IMPERSONATE permission for each user’s database account. This account acts on behalf of each user’s database account.

When the view displays, it is restricted by each user’s individual database permissions: Jane sees only the West Coast sales data, Henry sees only the East Coast data.

Impersonate with a Run As Service Account

Impersonating via a Run As service account is the recommended way to perform impersonation. The Run As service account is an Active Directory user account the Tableau Server service can run under on the machine hosting Tableau Server. This same account must have IMPERSONATE permission for the database user accounts in SQL Server.
From a data security standpoint, using the Tableau Server Run As service account for impersonation gives the administrator the most control.

To set up impersonation with a Run As User account:

1. Enable Kerberos Service Account Access.
2. Create a workbook in Tableau Desktop. When you create the data connection, select **Use Windows NT Integrated security** for the workbook’s live connection to a SQL Server database:

3. In Tableau Desktop, publish the workbook to Tableau Server (**Server > Publish Workbook**).
4. In the Publish dialog box, click Authentication, then in the Authentication dialog box, select **Impersonate via server Run As account** from the drop-down list:
5. Click OK.

6. Test the connection by signing into Tableau Server as a user. When you click a view, you should not be prompted for database credentials and you should only see the data the user is authorized to see.

Impersonate with Embedded SQL Credentials

You can also perform impersonation by having the person who publishes a view embed their SQL Server account credentials in the view. Tableau Server can be running under any type of account, but it will use these credentials, supplied by the publisher, to connect to the database.
This may be the right choice for your site if the account that handles the impersonation cannot be an Active Directory (AD) account and if you’re comfortable giving workbook publishers an account with a potentially high permission level on SQL Server.

**Note:**

To use this approach, **Embedded Credentials** must be enabled on the server Settings page in Tableau Server:

To impersonate with the workbook publisher’s SQL account:

1. In Tableau Desktop, create a workbook. When you create the data connection, select Use a specific username and password for the workbook’s live connection to a SQL Server database:

2. Publish the workbook to Tableau Server (**Server > Publish Workbook**).

3. In the Publish dialog box, click Authentication, then in the Authentication dialog box,
select **Impersonate via embedded password** from the drop-down list:

4. Click **OK**.

5. Test the connection by signing in to Tableau Server as a user. When you click a view, you should not be prompted for database credentials and you should only see the data the user is authorized to see.

**Authorization**

*Authorization* refers to how and what users can access on Tableau Server after authentication has been verified. Authorization includes:

- What users are allowed to do with content hosted on Tableau Server, including projects, sites, workbooks, and views.
What users are allowed to do with the data sources that are managed by Tableau Server.

What tasks users are allowed to perform to administer Tableau Server, such as configuring server settings, running command line tools, creating sites, and other tasks.

Authorization for these actions is managed by Tableau Server and determined by a combination of the user’s site role and permissions associated with specific entities such as workbooks and data sources.

Site roles

Site roles define who is an administrator. Administrators can be assigned at the site or server level. For non-admins, site roles indicate the maximum level of access a user can have on a given site, subject to permissions set on content assets. For example, if one user is assigned the Viewer site role, and another the Creator site role.

For more information about site roles, see Set Users’ Site Roles.

Permissions

Permissions determine whether a given user is allowed or denied to perform a specific action on a specific content asset.

As an administrator setting up Tableau Server, it’s important that you understand how permissions are evaluated. Understanding the Tableau permissions process will enable you to set up and configure permissions on sites, projects, and other assets so that you can control how content and data is shared, published, viewed, extracted, and imported.

Four important concepts to understand about permissions in Tableau are:

- **Permissions are asset-based.** Permissions are assigned to individual content assets (projects, data sources, workbooks) and are granted to users or groups.

- **Permissions are implicitly denied, and non-admin users must explicitly be allowed to access content.** The process by which Tableau Server determines the “allow” or “deny” permission is explained in detail in Evaluate permission rules.

- **Permissions inheritance exists only in locked projects and in workbooks with tabbed views.** When content permissions are locked to the top-level project,
workbooks, views, and data sources in the entire project hierarchy will use the default permissions set at the top-level project. In workbooks saved with the option **Show sheets as tabs**, views inside those workbooks use the workbook permissions. For more information, see Project permissions.

- **In a project that is not locked, initial permissions are a one-time copy of the container item’s permissions.** A data source or workbook starts with the default permissions, but authorized users can subsequently edit permissions on those assets. For more information on default permissions and projects, see Lock project permissions.

Tableau Server provides a flexible permissions infrastructure that allows you to manage access to all content for countless scenarios. For more detailed information, see Permissions.

### Data access and external authorization

There are scenarios where Tableau Server and Desktop rely on external authorization to enable access to data. For example:

- Users connecting to external data sources might require authorization that is outside the scope of Tableau Server’s authority. If users publish an external data source, Tableau Server will manage access and capabilities of that data source. But if users embed an external data source in a workbook, it’s up to the user who publishes the workbook to determine how other users who open the workbook will authenticate with the underlying data that the workbook connects to.

- Running Tableau Server in an organization with Active Directory, where Tableau has been configured with a Run As user account, results in a dependency on Active Directory and NTFS for authorization. For example, if you configure Tableau Server to use the Run As account to impersonate users connecting to SQL, then object-level authorization is reliant on NTFS and Active Directory.

- How users authenticate and are authorized by specific database solutions can differ. As noted, Tableau Server can be configured to provide access authorization when a data source is configured, but some databases will authorize access according to their own authentication scheme.
Data Security

Tableau provides several ways for you to control which users can see which data. For data sources that connect to live databases, you can also control whether users are prompted to provide database credentials when they click a published view. The following three options work together to achieve different results:

- **Database login account:** When you create a data source that connects to a live database, you choose between authenticating to the database through Windows NT or through the database’s built-in security mechanism.

- **Authentication mode:** When you publish a data source or a workbook with a live database connection, you can choose an Authentication mode. Which modes are available depends on what you choose above.

- **User filters:** You can set filters in a workbook or data source that control which data a person sees in a published view, based on their Tableau Server login account.

The table below outlines some dependencies with the above options:

<table>
<thead>
<tr>
<th>Database Connection Options</th>
<th>Data Security Questions</th>
</tr>
</thead>
<tbody>
<tr>
<td>Database login account uses...</td>
<td>Authentication mode</td>
</tr>
<tr>
<td>Active Directory credentials (Windows Authentication)</td>
<td>Kerberos service account</td>
</tr>
<tr>
<td></td>
<td>Impersonate via server Kerberos service account</td>
</tr>
<tr>
<td>Database Connection Options</td>
<td>Data Security Questions</td>
</tr>
<tr>
<td>----------------------------</td>
<td>-------------------------</td>
</tr>
<tr>
<td>Database login account uses...</td>
<td>Authentication mode</td>
</tr>
<tr>
<td>User name and password</td>
<td>Viewer enters their credentials</td>
</tr>
<tr>
<td></td>
<td>Prompt user: Viewers are prompted for their database credentials when they click a view. Credentials can be saved.</td>
</tr>
<tr>
<td></td>
<td>Embedded credentials: The workbook or data source publisher can embed their database credentials.</td>
</tr>
<tr>
<td></td>
<td>Impersonate via embedded password: Database credentials with impersonate permission are embedded.</td>
</tr>
</tbody>
</table>
Because it can create unexpected results, Tableau recommends that you not use this authentication mode with user filters.

User filters, the embedded credentials option and the impersonation modes have similar effects—when users click a view, they are not prompted for database credentials and they see only the data that pertains to them. However, user filters are applied in the workbook by authors, and the impersonation authentication modes rely on security policies defined by administrators in the database itself.

Best Practices for Row Level Security

Row Level Security (RLS) in Tableau refers to restricting the rows of data a certain user can see in a workbook. This differs from Tableau permissions, which are used to control access to content and feature functionality. For example, permissions control whether a user can comment on a workbook, while Row Level Security allows two users viewing the same dashboard to see only the data they are allowed to see.

For live connections and multi-table extracts, the basic Row Level Security workflow is:

1. The user is identified by logging into Tableau Server or Tableau Online
   - This requires a distinct username per user and secure single sign-on (SSO)
   - Active Directory, LDAP, or the Tableau Server REST API can be used to synchronize usernames and establish permissions
2. The set of data entitlements for the user is retrieved from all possible data entitlements
   - This requires a data structure that can link entitlements to the Tableau username
3. The data is filtered by the entitlements for that user
   - This often requires using user functions in a calculated field
4. The published, filtered data is used to build content
   - Using a published (rather than embedded) data source with a data source filter ensures the RLS cannot be modified by downloading or web editing the workbook

How the joins, calculated fields, and filters are set up depends on the structure of the data and how users are managed.
Entitlement tables

Any unique combination of attributes that the data can be filtered on is an entitlement. Most commonly, there are separate tables for specifying the entitlements themselves and mapping those entitlements to users or user roles. Denormalizing is recommended from a performance standpoint because joins are expensive operations.

The entitlements view, consisting of the entitlements mapped to users or roles, is joined with the data. A user-based data source filter is then applied, acting as a WHERE clause that brings in only the entitlements—and therefore the appropriate data rows—for the relevant user. (Query optimization should ensure the filtering occurs before joining when the query is processed to minimize data duplication. See Performance and processing order of operations for more information.)

Entitlement table models

Generally, there are two models for representing entitlements:

**Full mapping to the deepest level of granularity**

- Entitlements are defined fully for every column.
- There is one row in the mapping table for every possible entitlement the user has.
- This model requires fewer join clauses.

**Sparse entitlements**

- Entitlements are defined for every level of hierarchy, with NULL used to represent an “all” state.
- There is a single row in the mapping table for a particular level in the entitlement hierarchy, which vastly reduces the number of entitlement rows for users at high levels in a hierarchy.
- This model requires more complex joins and filters.
Users and roles

Combinations of entitlements are commonly represented as roles, which are then linked to users in a many-to-many mapping table. This allows for easily changing or removing a user from the role, while still maintaining a record of the role and its entitlements.

Alternatively, a many-to-many mapping table can be created that instead assigns users directly to entitlements as opposed to going through joining a role table. It will require managing the values more directly in the table but does eliminate a join.

**Note:** The user values associated with a role or entitlement need to match the username or full name on the Tableau site in order to take advantage of the user functions in Tableau Desktop.

Joins

Regardless of the model used to represent the entitlements, it is advisable to join all entitlements and mapping tables together into a single denormalized entitlements view. While at first this will cause a “blowup” (highly duplicative) version of the entitlements, the data source filter on the user will reduce it back down. You will also want this view if you plan on using an extract.

The deepest granularity method can have a performance benefit when everything is hierarchical—you only need to do a single join on the deepest level of the hierarchy. This only works if all of the attributes at the lowest level are distinct. If there is a chance for duplication (for example, a Central sub-region in more than one region), then you’ll need to join on all the columns to achieve the effect of a distinct key value.

The actual details and their performance characteristics depend on the data system and require testing. For example, using a single key could potentially improve the performance because the join is then only executing on one column, but correctly indexing all of the columns may give equal performance when other factors are taken into consideration.
Implement Row Level Security

Deepest granularity

Once the denormalized view of mapped entitlements is created, an inner join is set up between the view and the data in the Tableau data connection dialog. The data can remain in a traditional star schema. Alternatively, the dimension and fact tables can be materialized together into two views. Multi-table extracts will build extract tables to match the joins, so creating the two views will simplify the resulting extract. The SQL will follow this basic pattern:

```
SELECT *
FROM data d
INNER JOIN entitlements e ON
d.attribute_a = e.attribute_a AND
d.attribute_b = e.attribute_b AND ...
WHERE e.username = USERNAME()
```

Sparse entitlements

If your entitlements more closely resemble the sparse entitlements model, then the custom SQL to join the data to the entitlements would be a little more complex because of the NULL values. Conceptually, it would look like the following:

```
SELECT *
FROM data d
INNER JOIN entitlements e ON
(e.region_id = d.region_id OR ISNULL(e.region_id)) AND
(e.sub_region_id = d.sub_region_id OR ISNULL(e.sub_region_id)) AND
(e.country_id = d.country_id OR ISNULL(e.country_id))
```
Without using custom SQL, this can be done with a cross join and additional filters in Tableau Desktop. Create a join calculation on both sides of the join dialog that simply consists of the integer 1 and set them equal. This joins every row from the data table with every row in the entitlements table.

Then you need a calculation (or individual calculations) to account for the levels in the hierarchy. For example, you could have several calculations that follow this format: 

\[
\text{[region_id]} = \text{[region_id (Entitlements View)]} \text{ OR ISNULL([region_id (Entitlements View)]}
\]

Or you could have a combined calculation for all levels in one:

\[
([\text{region_id}]=\text{[region_id (Entitlements View)]} \text{ OR ISNULL ([region_id (Entitlements View)]}) \text{ AND }
([\text{sub_region_id}]=\text{[sub_region_id (Entitlements View)]} \text{ OR ISNULL ([sub_region_id (Entitlements View)]}) \text{ AND }
([\text{country_id}]=\text{[country_id (Entitlements View)]} \text{ OR ISNULL ([country_id (Entitlements View)]})
\]

The ISNULL function matches any entitlement column to all items in the other column. As always with RLS, these calculations should be added as data source filters.

Data source filter

For both approaches, once the entitlements are correctly joined with the data, a filter needs to be set up to limit the data for a specific user. A calculated field should be created with a user function. For example, a simple Boolean comparison of whether the user listed in the Username field is the same as the username of the person logged into the Tableau site:

\[
[\text{Username}] = \text{USERNAME()}
\]

This calculation should be used as a data source filter (with TRUE selected).
If the data source is embedded and a user has permissions to web edit or download the workbook, then the RLS is nonexistent since the filters enforcing it can be easily removed. The Tableau data source should be published separately as opposed to being left embedded in the workbook.

All access with deepest granularity

There is also a common scenario in which there are two access levels within the organization: people who can see everything (“all access”) or people with some reasonably definable subset of entitlements. This is most commonly seen for embedded applications—the organization hosting the data can see everything, but each client can only see their own data. In this case, you need a way to return the full data for the “all access” users, while maintaining the deepest granularity joins for all other users.

For this technique, you will use Tableau groups to create an override using a calculation in the join condition.

1. Create a group for users who should see all the data (here called All Access)
2. From the fact view, create a left join with two join conditions
   - The first join condition should be on the column that represents the deepest level of granularity
   - The second join condition should be two calculations:
     - On the left side (the fact view), for the calculation, enter True
     - On the right side (the entitlements view), the calculation should be: IF ISMEMBEROF('All Access') THEN False ELSE True END
3. On a sheet, create a calculation structured as: [Username] = USERNAME() OR ISNULL([deepest granularity column] (Entitlements View))
4. Create a data source filter on the username calculation

If a user is a member of the All Access group, then the join becomes a left join on True = False. This means there are no matches at all in the entitlements view, so the entire fact view is returned with NULLs for the columns from the entitlements view (zero duplication). In the case where the user is not part of the All Access group, the True = True join condition doesn’t change anything and the join will function as expected.
The user calculation used as a data source filter is true for all rows when the group override is working, or it will filter down to only the user’s deepest granularity in the hierarchy.

Performance and processing order of operations

When a visualization is viewed in Tableau (Desktop, Server, or Online), Tableau sends an optimized query to the RDBMS which then processes the query and sends results back to Tableau to render the visualization with the resulting data. The order of operations for when joins, calculations, and filters are carried out depends on the query optimizer and how the query is executed.

Live connections

When using a live connection to a data source in Tableau, the performance of the query execution is dependent on the query optimizer which translates the incoming SQL into an efficient plan for retrieving the data.

There are two ways the query can be processed:

1. Filter the entitlement rows to the user then join to the fact table
2. Join the entitlements to the fact table then filter to the user’s rows

In an ideal situation, the query optimizer will ensure the database processes the query by filtering then joining. If a user is entitled to everything, this means the maximum number of rows processed will be the number of rows in the data table.

If the database processes the query by joining then filtering, there may be duplication of data. The maximum number of rows processed will be the number of users entitled to see that particular row times each row in the data table.

It will be clear if this second scenario happens: your queries take a long time to finish, you get errors, or there is an indication of performance issues in the database. Your total data volume will expand exponentially, which could cause inordinate system strain on the backend.
Extracts

When the data source in Tableau is a live connection, Tableau sends every query that is necessary to render a particular viz or dashboard to the RDBMS. When the data source is an extract, the process of querying data from the underlying data source only happens at extract creation and refresh. All of the individual queries for visualizations are answered by the extract engine from the extract file.

The same order of operations issue is present when building single table extracts. However, the “blowup” will happen both on the underlying data source and within the resulting extract itself.

Considerations with extracts

Starting in version 2018.3, the data engine can create a multi-table extract and RLS can be implemented as described above. Using multiple table extracts reduces the time it takes to generate an extract with many-to-many relationships by not materializing the join.

The extract should be built with a data object and an entitlements object. This is the simplest storage in the extract and results in the best performance.

- The data object is the table, view or custom SQL query that represents the denormalized combination of the fact and necessary dimension tables
- The entitlements object is a denormalized table, view or custom SQL query of whatever entitlements are necessary to filter the data at the most granular level, which requires:
  - A column for user name matching the exact usernames in Tableau Server or Tableau Online
  - A row for each of the most granular entitlements to the data object

This format is laid out in the deepest granularity method above. Multi-table extracts use the same method, with the caveat that only two data objects are being joined and any field-specific filtering is already applied within the object.
Because multiple table extracts have extract filters disabled, you can filter either in the views or tables you connect to in the data source, or define the filters in custom SQL objects in the Tableau data connection dialog.

**Note:** As with live connections, if the data source is embedded and a user has permissions to web edit or download the workbook, then the RLS is nonexistent since the filters enforcing it can be easily removed. The extract should be published separately as opposed to being left embedded in the workbook.

Single table extracts

The following method is only recommended when using a version of Tableau prior to 2018.3—multiple table extracts are preferable if available.

Single table extracts materialize any joins you build when constructing the Tableau data source and stores everything as a single table through one query, the results of which are transformed in a single table in the extract file. This denormalization carries the risk of causing massive data duplication, as every row that was allocated to more than one entitlement or user would be duplicated as a result of the many-to-many relationship.

To prevent this duplication:

1. Create a Security Users Field that contains the user names for that entitlement
   - for example, a value may be “bhowell|mosterheld|rdugger”
2. Use the CONTAINS() function within Tableau to correctly identify individual users
   - For example, CONTAINS([Security Users Field], USERNAME())

This method obviously has some caveats. It requires that you go from your entitlements in rows to a single column separated correctly using SQL, and that column can only contain so many characters. Partial matches can be trouble, and you need to use separators that will never be valid in the IDs themselves. Although it is performant within the Tableau Data Engine, as a string calculation it will be very slow for most databases. This limits your ability to switch back to a live connection.
Alternatively, you can take different extracts per “role” or entitlement level, so that only the data appropriate to that person or level is contained within the extract, but this will require processes to appropriately permission and leverage template publication within Tableau Server, generally via the APIs.

Use built-in Row Level Security in a data source

Many data sources have mechanisms for RLS built in. If your organization has already put effort into building Row Level Security in a data source, you may be able to take advantage of your existing RLS. It is not necessarily easier or better to implement a built-in RLS model vs. building it with Tableau in mind; these techniques are generally leveraged when an organization has already invested in these technologies and they want to take advantage of the investment. The main benefit of using built-in RLS is that administrators can implement and control their data security policy in one place: their databases. For more information, see Row Level Security in the Data Source.

Row Level Security in the Data Source

If your organization has already put effort into building out Row Level Security (RLS) in a data source, you may be able to use one of the following techniques to take advantage of your existing RLS. In order to leverage the data source’s security models, live connections are required. Additionally, these techniques are likely not available in Tableau Online; the Tableau username for Online is a unique email address which is not typically the user identity on the data source side.

It is not necessarily easier or better to implement a built-in RLS model vs. building it with Tableau in mind; these techniques are generally leveraged when an organization has already invested in these technologies and they want to take advantage of the investment.

Impersonation (Microsoft SQL Server)

Microsoft SQL Server (and a few related systems) can be configured so that users of the database only have access to views with RLS filters built in, either utilizing Security Junction Tables or views built by the DBA. Tableau can take advantage of this using a concept called “impersonation.”
When publishing a Tableau Data Source containing an MS SQL Server connection to Tableau Server, there are two authentication options available to take advantage of impersonation. The menu you see will depend on whether you logged into the SQL Server using network authentication or by entering username/password credentials.

To enable RLS filtering for any user who can access the Published Data Source in Tableau Server, either the AD Run-As Account or the embedded SQL server credentials must have permission to EXECUTE AS for all of the Tableau users in the database that will be accessing the dashboard or data source. All Tableau users must exist in the database server as users, with SELECT rights for the Views you are trying to connect to (and have RLS applied to). See Impersonation Requirements for the comprehensive list of requirements.

Kerberos & constrained delegation

Constrained delegation within Tableau Server using Kerberos operates similarly to impersonation in that it allows Tableau Server to use the Kerberos credentials of the view of a workbook or view to execute a query on behalf of the viewer, so if RLS is set up on the data source, the viewer of the workbook will only see their data.

To see the comprehensive list of data sources where Kerberos delegation is supported, see Enable Kerberos Delegation. Active Directory is required; the computer where Tableau Server is installed must be joined to the Active Directory domain. The authentication method specified when publishing the data source must be viewer credentials.

Note that Kerberos can be leveraged for RLS when using Microsoft Analysis Services.

OLAP Cubes

OLAP Cube connections in Tableau do not have the equivalent of a data source filter, which is required for the entitlements table-based RLS method in Tableau, or access to the USERNAME() function. For these reasons, Kerberos and constrained delegation is a recommended approach to RLS with OLAP data sources, which allows Tableau to leverage user filtering that has already been implemented on the OLAP Server side.

If the users viewing the dashboard will not be part of the domain, then the manual approach to creating user filters is possible. However, because the User Filter Set generated cannot
be added as a data source filter, and will instead exist on the filters shelf, it is important that Web Editing and Download Workbook functionality is not permissible for any published views utilizing this method.

SAML delegation & SAP HANA

If Tableau Server is configured to use Configure SAP HANA SSO to provide a single sign-on experience, the viewer credentials are used to execute the query as that user, which will operate within whatever security is applied on the user level. The authentication method specified when publishing the data source must be viewer credentials.

Initial SQL to force a user-specific session (Oracle VPD)

Initial SQL allows you to specify a SQL command that is run when the connection is made to the database for the purpose of setting up temporary tables to use during the session or to set up a custom data environment.

For Oracle VPD, you can set up a session specific to a user by running a particular stored procedure or function to set the context of the database connection to match the Tableau user’s username:

begin
  DBMS_SESSION.SET_IDENTIFIER([TableauServerUser]);
end;

The same high-level requirements hold true for using this for RLS as with impersonation; the DBA must set up VPD and all of the associated users to exist on the database.

On MS SQL Server, you could force an EXECUTE as command (however, this is similar to what Tableau does with impersonation already):

EXECUTE AS USER = [TableauServerUser] WITH NO REVERT;
**Note:** If the data source is embedded and a user has permissions to web edit or download the workbook, then the RLS is nonexistent since the initial SQL enforcing it can be easily removed. The data source should be published separately as opposed to being left embedded in the workbook.

### Comparison matrix for Row Level Security Methods

<table>
<thead>
<tr>
<th>Method</th>
<th>Useful when</th>
<th>Pros</th>
<th>Cons</th>
</tr>
</thead>
</table>
| Entitlements table (Recommended) | • There is an existing concept of entitlements in the database  
• The organization is setting up Row Level Security for the first time | • Easy to test, update, maintain, and scale  
• Works for both live connections and extracts in version 2018.3+ | • Requires creating and maintaining entitlements table  
   • Could require selecting and creating appropriate keys to optimize for performance |
| CONTAINS() with extracts       | • Implementing RLS in extracts prior to version 2018.3           | • Allows you to take advantage of extract efficiencies              | • Requires mapping all users to a single column  
   • Difficult to switch back to live connections because of string calculation |
| Impersonation                 | • Every user accessing the                                       | • Security is handled and                                           | • Requires every person access- |
data will exist as a user in your SQL server (Usually, internal deployments) maintained in one place—the database

Kerberos
- All necessary data sources are set up for Kerberos delegation and RLS is set up on the data source (usually internal deployments)
- The viewer’s name appears on the access logs for the data source
- Security is handled and maintained in the data source
- Tableau must be configured to use LDAP-Active Directory
- Tableau Server must be joined to the AD domain
- Every user must exist within your AD domain

Initial SQL
- The database supports Initial SQL and RLS is set up on the data source side
- Allows the passing of Tableau parameters at load time
- Dedicated connection that can’t be shared with other users
- Users must exist within data source to execute query
- Not all data sources support Initial SQL
- Potential performance implications because of restricted cache sharing
Manage Server Secrets

Tableau Server needs to store a number of secrets it uses to perform various functions, typically securing internal communication, communicating with other applications or the operating system, or providing secure communication with clients. In this context, the term secret may refer to a password, a token, or other string that is used to authenticate one entity to another.

There are two categories of secrets that are required to run Tableau Server. They differ according to how the secrets are generated:

- **Secrets that are generated by administrators.** These include credentials and associated secrets for the Run As User account and the SMTP credentials used by Tableau Server.
- **Secrets that are automatically generated by various processes in the system.** For example, a secret is required to protect communication between the Cluster Controller and ZooKeeper processes. And a number of different passwords are required for each service and programmatic user that communicates with Postgres.

Most secrets are encrypted while at rest. When a secret is needed, it is decrypted at run time.

This topic describes how secrets storage works and what you need to do to properly manage storage of secrets on Tableau Server.

Understanding how secrets storage works

During installation Tableau Server generates and stores a master key in a Java keystore. The master key is used to encrypt a configuration encryption key that is used across the system.

Whenever a new secret is created or updated, the secret is encrypted with the configuration encryption key. The encrypted value is then stored with its corresponding configuration parameter in a YAML file on the server. Parameters that hold an encrypted value use the format,
ENC(<encrypted string>), where <encrypted string> is a Base64-encoded encrypted string.

At run time, when a given secret needs to be accessed, the encrypted values are read into memory and decrypted with the configuration encryption key.

In the case of pending changes, where secrets are entered during a configuration change, the entire transaction is encrypted. In this scenario, after you enter a secret and then save the pending change, the secret is transmitted to the Coordination Service (over encrypted SSL). The Coordination Service encrypts the secret and stores it until the pending changes are applied. When changes are applied, the secret (still encrypted) is promoted to the current configuration version.

Tableau Server encrypts secrets using 256-bit AES in GCM mode. The keys used for secure storage are different than the asset keys that are used to encrypt embedded database credentials before they are stored in the repository.

Who has access to the master key?

In a default installation, the Java keystore for Tableau Server will be replicated into the /tabsvc/keystores folder for each service on that node under /var/-opt/tableau/tableau_server/data/tabsvc/config.

For example,

/var/opt/tableau/tableau_server-
data/tabsvc/config/tabadminagent_<version_number>/tabsvc/keystores/tableauserver.jks.

If you use a custom install directory, then the keystore files will be found under

<install directory>/tableau_server/data/tabsvc/config/<service name_.version_number>/tabsvc/keystores

By default, the following users and groups have access to this directory:
Tableau Server on Linux Administrator Guide

- root
- tableau (user)
- members of the 'tableau' group

Import and export configuration information

Tableau Services Manager introduces the capability to import and export configuration information using tsm configuration.

**Note:** This version of Tableau Server does not support restoring configuration information from a backup. Instead, we recommend using the export and import configuration commands to backup and restore configuration information.

While configuration secrets are encrypted when stored on disk internally, when the configuration is exported to a file, secrets are written into the file in plain text. It is up to the administrator to take measures to protect this file. There are a variety of options available:

- Write the file to an encrypted file system.
- Write the file to a directory that is restricted to specific users or groups by file system permissions.
- Encrypt the output file.

Securing secrets for import and export operations

This section describes how to PGP encrypt the backup output. With this method, you will create a named pipe and provide it as the file argument, then use that as input to gpg encrypt it. The advantage is the secrets are never written to disk in plain text. gpg must be available and a private key available. An example of this method is shown in the following section.

The example in this section describes one way of handing secrets to store them on a separate computer.

See the following external references for more information:

- File encryption on the command line
- mkfifo(1) - Linux man page
Example: encrypt and export

The following is an example of how to secure the file when exporting the configuration.

```sh
mkfifo -m 600 /tmp/secure1 && (gpg --symmetric --batch --yes --passphrase-file ~/.secrets/pgppassphrase.txt --cipher-algo AES256 --output encrypted.enc < /tmp/secure1 &) && tsm settings export -f /tmp/secure1 && rm /tmp/secure1
```

The details of this operation are:

- Create a named pipe with access limited by file permissions to rw for current user.

  ```sh
  mkfifo -m 600 /tmp/secure1
  ```

- Call gpg to encrypt the data sent to the named pipe, backgrounding it to a separate process. It will block waiting for data. The result will be a file containing the encrypted data.

  ```sh
  gpg --symmetric --batch --yes --passphrase-file ~/.secrets/pgppassphrase.txt --cipher-algo AES256 --output encrypted.enc < /tmp/secure1 &
  ```

- Call tsm to export the configuration, providing the named pipe as the file argument.

  ```sh
  tsm settings export -f /tmp/secure1
  ```

- Delete the named pipe.

  ```sh
  rm /tmp/secure1
  ```

The encrypted data is in the file "encrypted.enc."

Example: decrypt and import

The following is an example of how to decrypt and import the configuration.
Tableau Server on Linux Administrator Guide

```bash
mkfifo -m 600 /tmp/secret2 && (gpg --decrypt --batch --yes --passphrase-file ~/.secrets/pgppassphrase.txt encrypted.enc > /tmp/secret2 &)
&& tsm settings import -f /tmp/secret2 && rm /tmp/secret2
```

The details of this operation are:

- Create a named pipe with access limited by file permissions to rw for current user.

  ```bash
  mkfifo -m 600 /tmp/secure2
  ```

- Decrypt the configuration and send it to the named pipe. Background this to a separate process, it will block waiting to be read.

  ```bash
  gpg --decrypt --batch --yes --passphrase-file ~/.secrets/pgppassphrase.txt encrypted.enc > /tmp/secret2 &
  ```

- Execute the tsm configuration import command, logging in as needed.

  ```bash
  tsm settings import -f /tmp/secret2
  ```

- Delete the named pipe.

  ```bash
  rm /tmp/secure1
  ```

The pending configuration contains the imported configuration.

Run tsm pending-changes apply to commit changes. The `pending-changes apply` command displays a prompt to let you know this will restart Tableau Server if the server is running. The prompt displays even if the server is stopped, but in that case there is no restart. You can suppress the prompt using the `--ignore-prompt` option, but this does not change the restart behavior. For more information, see tsm pending-changes apply.

Cluster nodes

When adding a new node to your Tableau Server cluster, you will first need to generate the node configuration file (tsm topology). The node configuration file contains a copy of the
master keystore file used for encrypting the configuration secrets.

**Important:** We strongly recommend that you take additional measures to secure the node configuration file when exporting a configuration file with secrets.

When installing and configuring Tableau Server on the new node, you will need to provide the node configuration file to the `initialize-tsm` command. You can use a similar technique as described above to decrypt the contents of the file that was previously encrypted and send it via a named pipe to the `initialize-tsm` command.

Secrets storage event logging

The following events related to secrets storage are logged:

- Generating new encryption keys
- Encryption key is rolled or changed
- Encrypting a new value in the configuration file

For more information about log files and where they are stored, see Work with Log Files.

Managing secrets

As a Tableau Server administrator the most important task related to secrets storage is to periodically update secrets. In some cases (server troubleshooting or auditing), you may need to retrieve a password.

For other operations, such as upgrading versions, backing up and restoring, or adding new nodes to a cluster—as noted above—Tableau Server manages secrets storage and related processes automatically.

Updating secrets

You should update secrets periodically, according to your company’s security policy.

To update the master key and automatically generated secrets, run `tsm security regenerate-internal-tokens`. 
Retrieving passwords

In some cases, you may need to retrieve a password for troubleshooting or other operations. For example, you may need the Postgres readonly user credentials that are generated and encrypted by Tableau Server. In these cases, you can run a tsm command that will retrieve and decrypt the password for you.

To retrieve a password, open Command Prompt and issue a `tsm configuration get` command for one of the parameters listed in the table below.

For example, to retrieve a password for the readonly Postgres user, type the following command:

```
tsm configuration get -k pgsql.readonly_password
```

The command will return the password in clear text:

```
$ tsm configuration get pgsql.readonly_password
password
```

<table>
<thead>
<tr>
<th>Configuration Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>clustercontroller.zookeeper.password</td>
<td>Password for cluster controller to connect to zookeeper.</td>
</tr>
<tr>
<td>elasticserver.client.password</td>
<td>Password for logging into Elastic search service.</td>
</tr>
<tr>
<td>elasticserver.ssl.admin.cert.bytes</td>
<td>Admin certificate that is used for administrative access to the Elastic search service. The admin certificate is used to generate the node certificate.</td>
</tr>
<tr>
<td>elasticserver.ssl.admin.key_file_bytes</td>
<td>Certificate key for administrative access to the Elastic search service.</td>
</tr>
<tr>
<td>elasticserver.ssl.node.cert.bytes</td>
<td>Certificate that is used for Elastic node-to-node communication.</td>
</tr>
<tr>
<td>Variable</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>elasticserver.ssl.node.key.file_bytes</td>
<td>Certificate key that is used for Elastic node-to-node communication.</td>
</tr>
<tr>
<td>elasticserver.ssl.root.cert.bytes</td>
<td>Certificate that is used to sign the admin and node certificates. This certificate is used by TSM for health check and by NLP to connect to Elasticserver.</td>
</tr>
<tr>
<td>elasticserver.ssl.root.key.file_bytes</td>
<td>Certificate key for root certificate.</td>
</tr>
<tr>
<td>filestore.zookeeper.password</td>
<td>Password for filestore to connect to zookeeper.</td>
</tr>
<tr>
<td>hyper.connection.init_password</td>
<td>Password used to initialize the Hyper database for user <code>tableau_internal_user</code> and is then used for connecting to Hyper.</td>
</tr>
<tr>
<td>jdbc.password</td>
<td>Password for the rails Postgres user.</td>
</tr>
<tr>
<td>kms.persistent_store</td>
<td>A collection of master encryption keys (MEKs) used by the Key Management System.</td>
</tr>
<tr>
<td>maestro.rserve.password</td>
<td>Password for connecting to an external Rserve instance used by Tableau Prep Conductor for running flows that have nodes with R scripts.</td>
</tr>
<tr>
<td>maestro.tabpy.password</td>
<td>Password for connecting to an external TabPy (Python server) instance used by Tableau Prep Conductor for running flows that have nodes with Python scripts.</td>
</tr>
<tr>
<td>oauth.google.client_secret</td>
<td>Client secret of the Google Cloud Platform account.</td>
</tr>
<tr>
<td>oauth.quickbooks.consumer_secret</td>
<td>Consumer secret of the Intuit developer</td>
</tr>
<tr>
<td>Environment Variable</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------</td>
<td>-------------</td>
</tr>
<tr>
<td>oauth.salesforce.client_secret</td>
<td>Client secret of the Salesforce developer account.</td>
</tr>
<tr>
<td>pgsq1.adminpassword</td>
<td>Password for the tblwgadmin Postgres user.</td>
</tr>
<tr>
<td>Note: Although the configuration parameter is encrypted in Tableau's configuration files (tabsvc.yml, workgroup.yml), this password is stored in plain text in other files used by SAML and the Postgres recovery process.</td>
<td></td>
</tr>
<tr>
<td>pgsq1.readonly_password</td>
<td>Password for the readonly Postgres user.</td>
</tr>
<tr>
<td>pgsq1.remote_password</td>
<td>Password for the tableau Postgres user.</td>
</tr>
<tr>
<td>redis.password</td>
<td></td>
</tr>
<tr>
<td>searchserver.client.password</td>
<td></td>
</tr>
<tr>
<td>servercrashupload.proxy_server_password</td>
<td>Password for custom proxy server used to upload crash reports.</td>
</tr>
<tr>
<td>service.runas.password</td>
<td>Password of the Run As users. Stored temporarily.</td>
</tr>
<tr>
<td>ssl.cert.file_bytes</td>
<td>The content of one of the three SSL certificate files uploaded by the administrator. The certificate files are required to enable secure external connections to Tableau Server.</td>
</tr>
<tr>
<td>Variable Name</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>ssl.chain.file_bytes</td>
<td>The chain file(s) for the certificates uploaded by the administrator for external SSL.</td>
</tr>
<tr>
<td>ssl.key.file_bytes</td>
<td>Key file(s) for the certificates uploaded by the administrator for external SSL.</td>
</tr>
<tr>
<td>ssl.key.passphrase</td>
<td>Optional passphrase used to protect the external SSL key.</td>
</tr>
<tr>
<td>svcmonitor.notification.smtp.password</td>
<td>SMTP Server password supplied by the administrator through TabConfig.exe.</td>
</tr>
<tr>
<td>tabadminservice.password</td>
<td>Password for the service that allows server admins to download log files through the web interface.</td>
</tr>
<tr>
<td>vizportal.openid.client_secret</td>
<td>This is the password (&quot;provider client secret&quot;) used for OpenID Connect SSO.</td>
</tr>
<tr>
<td>vizqlserver.external_proxy_password</td>
<td>Password used to authenticate to an external proxy.</td>
</tr>
<tr>
<td>vizqlserver.extsvc.password</td>
<td>Password for the service that supports R functionality in workbooks.</td>
</tr>
<tr>
<td>vizqlserver.rserve.password</td>
<td>Password for connecting Tableau Server to an external Rserve or TabPy instance.</td>
</tr>
<tr>
<td>wgserver.domain.password</td>
<td>Password used to bind to Active Directory.</td>
</tr>
<tr>
<td>wgserver.saml.key.passphrase</td>
<td>Passphrase used to access the PKCS#8 SAML key file.</td>
</tr>
<tr>
<td>zookeeper.tsm.password</td>
<td>Password that TSM uses to connect to Zookeeper coordination service</td>
</tr>
</tbody>
</table>
Extension Security - Best Practices for Deployment

The following information is for IT officers and administrators, Tableau server and site administrators, and anyone who is interested in managing dashboard extensions and the security of their data and business. The suggestions for deployment are intended for companies that have a mix of users who are on Tableau Desktop and Tableau Server or Tableau Online.

Security for extensions in Tableau

Extensions are web applications that could be hosted inside your network, or outside on a third-party server, or in a secure sandboxed environment hosted by Tableau. Extensions can interact with other components in the dashboard and potentially have access to the visible and underlying data in the workbook (through a well-defined API).

Sandboxed and Network-enabled extensions

Tableau supports two types of extensions: Network-enabled extensions, which could be hosted on web servers that are located inside or outside of your local network and have full access to the web, and Sandboxed extensions, which run in a protected environment without access to any other resource or service on the web.

Sandboxed extensions are hosted by Tableau and provide the most security and can eliminate certain risks like data exfiltration. Network-enabled extensions have full access to the web, which means that while they can offer rich features and experiences by being able to connect to outside resources, they should be carefully evaluated before deploying or adopting.
You can use Sandboxed and Network-enabled extensions in Tableau Desktop, Tableau Server, and Tableau Online. Tableau Server and Tableau Online provide the most control over the extensions your users can run.

Potential security risks with Network-enabled extensions

Because extensions are web applications there is the potential that a Network-enabled extension could be vulnerable to certain types of malicious attacks, which in turn could present a risk to your computer or data. The Open Web Application Security Project (OWASP) annually identifies the most critical web application security risks. These risks include the following:

- SQL injection
- Cross-site scripting (XSS)
- Sensitive data exposure

These risks could compromise the extension if the developers of the extension do not properly validate and handle user inputs, or if they generate dynamic queries to access sensitive databases. As you evaluate the extensions that you want to allow in Tableau, be sure to con-
sider how they manage authentication, data access, or user input, and how they mitigate security risks.

Mitigating the security threats with Network-enabled extensions

Understanding what an extension does is a first step to identifying the risks for your enterprise. In many cases, a dashboard extension does not access underlying data in the workbook and all the JavaScript code runs in the context of the browser running on the user’s computer. In these cases, no data leaves the computer even though the extension might be hosted on a third-party site outside of your domain. Some extensions allow you to connect Tableau with other applications you have already deployed in your domain.

Tableau provides security measures and security requirements for extensions. These are enabled for Tableau Desktop, Tableau Server, and Tableau Online.

- All extensions must use the HTTP Secure (HTTPS) protocol.
- By default, anyone using a dashboard with a Network-enabled extension will be prompted and asked to allow or deny the extension permission to run. The extension must request permission if it will access underlying data.
- To run on Tableau Server or Tableau Online, the URL of the Network-enabled extension must be added to a safe list. The site administrator manages this list.
- On Tableau Server and Tableau Online, the site administrator can control whether the prompt appears for each Network-enabled extension.

For more information, see Manage Dashboard Extensions in Tableau Server.

Manage extensions using Tableau

Extensions provide a way to add unique features to dashboards. You can use extensions to directly integrate the dashboard with applications outside of Tableau. While extensions open up a world of possibilities, there are instances where you need or want to maintain control of how extensions are deployed in your company or enterprise. In this respect, extensions are no different from any other software that you intend to use. Before you deploy software applications in your company you should thoroughly test and verify that the software works as expected and is secure. The same is true for extensions.
After you determine what level of access your users should have, and identify the extensions you want to use (or conversely, the extensions you don’t want used), you can use the controls and features within Tableau to restrict and curate the dashboard extensions users have access to.

- Do you need to restrict who can add or use extensions in Tableau Desktop? See Recommendations for Tableau Desktop
- Do you need to restrict or control the extensions your users have access to? See Recommendations for Tableau Server and Tableau Online.

Recommendations for Tableau Desktop

You have a range of options for deploying Tableau Desktop in your company. You can allow unrestricted access to Sandboxed and Network-enable extensions, or you can put limits and restrictions on who has access to extensions and under what circumstances.

- **All users** - All Tableau Desktop users are empowered to use extensions. If you want to enable your Tableau Desktop users to choose which extensions they should run, you can use the default installation settings. Extensions are enabled by default. Tableau Desktop users have unrestricted access to Sandboxed and Network-enabled extensions. If you need to place restrictions because of security concerns over Network-enabled extensions, you can disable access to Network-enabled extensions, but still allow all users to select Sandboxed extensions.
- **Select users** - Only some Tableau Desktop users are empowered to use extensions. Others can use web authoring to access extensions, as determined by server and site administrators. If you want to limit who in your organization has access to extensions, and only let those Tableau Desktop users choose which extensions they should run, you can use the custom installation settings and use that to turn off extensions by default for most users. You can selectively enable extensions for the Tableau Desktop users. You can further customize the options by allowing some users access to Sandboxed extensions only, and others to have access to both Sandboxed and Network-enabled extensions.
- **Web authoring only** - Only allow users to add extensions on Tableau Server and Tableau Online. To provide you with the most control over which extensions users have access to, you can turn off extensions for all Tableau Desktop users and then allow users to add extensions to dashboards on Tableau Server or Tableau Online using web authoring. In this scenario, the server and site administrators can
determine which extensions to allow users access to. Administrators can use the server and site settings to restrict access to Sandboxed extensions only, or to restrict access to Sandboxed extensions and the Network-enabled extensions that have been added to a safe list.

Selectively enable extensions on Tableau Desktop

By default, Tableau Desktop allows extensions and Tableau Desktop users have unrestricted access to Sandboxed and Network-enabled extensions. You can use options during installation to change the default setting and turn off support for all extensions, or turn off support for Network-enabled extensions. You can also turn off extensions after installation using registry settings and scripts.

- During Tableau Desktop installation: You can turn support for all extensions off, using the `DISABLEEXTENSIONS` property (`DISABLEEXTENSIONS=1`). See Install Tableau Desktop from the Command Line.

- During Tableau Desktop installation: You can turn Network-enabled extensions off, using the `DISABLENETWORKEXTENSIONS` property (`DISABLENETWORKEXTENSIONS=1`). If extensions are enabled, this setting allows only Sandboxed extensions. See Install Tableau Desktop from the Command Line.

- After Tableau Desktop installation: You can turn off support for all extensions, or just Network-enabled extensions, by editing the Registry (Windows) or running a script (Mac) on each Desktop. See Turn off dashboard extensions.
Recommendations for Tableau Server and Tableau Online

If your users have access to Tableau Server or Tableau Online, you can use the built-in security controls to put limits and restrictions on the extensions that can be used and under what circumstances. If you have turned off extensions on Tableau Desktop, you can still allow users to add extensions in web authoring, but you can limit the number of extensions that can be used to just ones you approve of.

![Extensions for Tableau Dashboards](image)

Trust Sandboxed extensions and the Network-enabled extensions on the safe list

Starting with Tableau 2019.4, only Sandboxed extensions are allowed to run by default, and no Network-enabled extensions are allowed to run unless they have been added to the safe list. Users can only use Sandboxed extensions and the Network-enabled extensions that are explicitly listed on the settings page for the site (Settings > Extensions > Enable specific Extensions).
**Note** To make the safe list the default behavior in Tableau 2018.2 and Tableau 2018.3, you need to change the settings for the site. On the Extensions settings page, under **Default behavior for Extensions**, clear the **Enable unknown extensions**... option.

In Tableau Server 2019.1, Tableau 2019.2, and Tableau 2019.3, no extensions are allowed to run unless they have been added to the safe list.

**Checklist for the safe list:**

- Does the extension come from a source that you know and trust?
- Check the URL of the extension. Does the URL look suspicious or contain dubious domain names?
- Does the extension require access to full (underlying data) or summary data? See [Understand data access](#).
- Test the extensions before allowing broad use. See [Test extensions for security](#). See [Test Network-enabled extensions for security](#).

**Add extensions to the safe list:**

- See [Add extensions to the safe list and configure user prompts](#).

**Block specific extensions from running on Tableau Server**

On Tableau Server, you can block specific extensions by adding their URL to the block list. This is useful if you have multiple sites that are configured differently for extensions. For example, if you have a test site where you want to be able to test internal or third-party extensions, you might have enabled the default behavior for extensions, where unlisted extensions are allowed to run provided they do not access the underlying data in the workbook. Adding an extension to the block list will prevent it from inadvertently being used on the test site.

- Add the URL of the extensions that you do not want to allow to the blocked list. This option is only available on Tableau Server. See [Block specific extensions](#).
Turn off extensions for a site

By default, extensions are enabled on Tableau Server and Tableau Online. On both Tableau Server and Tableau Online, the site administrator can turn off extensions for the site. The server administrator can turn off extensions for Tableau Server, this overrides the site settings. You should not have to change this setting. Only the extensions you add to the safe list are allowed. If you don’t add the URL of the extension to the safe list, the extension will not run.

- To disable extensions on a site (Tableau Server, Tableau Online), change the site settings that enables users to run extensions on the site. See Control dashboard extensions and access to data.

Show or hide user prompts to run Network-enabled extensions

When you add a Network-enabled extension to the safe list, you can configure whether users see prompts by default when they are adding the extension to a dashboard, or when they are interacting with a view that has the extension. The prompt tells users details about the Network-enabled extension and whether the extension has access to full data. The prompt gives users the ability to allow or deny the extension from running. You can hide this prompt from users, allowing the extension to run immediately. When enabled for a site, Sandboxed extensions are allowed by default and do not prompt users.

Turn off Sandboxed extensions

Starting in Tableau 2019.4, Sandboxed extensions are enabled for Tableau Server and Tableau Online by default. Sandboxed extensions run in a protected environment and are hosted by Tableau. Administrators can control whether to let users run Sandboxed extensions on a site. Sandboxed extensions don’t need to be added to the safe list. When Sandboxed extensions are allowed, users are able to freely add Sandboxed extensions to dashboards and are able to open and use dashboards that contain Sandboxed-extensions. If you need to block a Sandboxed extension, a server administrator can add the Sandboxed extension to a global block list. If you need to turn off Sandboxed extensions completely, you can change the default setting for the site.
Extract Encryption at Rest

Extract encryption at rest is a data security feature that allows you to encrypt .hyper extracts while they are stored on Tableau Server.

Tableau Server administrators can enforce encryption of all extracts on their site or allow users to specify to encrypt all extracts associated with particular published workbooks or data sources.

Limitations

Before they can be encrypted, older .tde file extracts must be upgraded to .hyper file extracts. This happens automatically as a part of the encryption job. For more information about the impacts of extract upgrade, see Extract Upgrade to .hyper Format.

Temporary files and cache files are not encrypted at rest with this feature.

Workbooks (.twb) and data source files (.tds) are not encrypted with this feature. These files will contain metadata such as database table column names and formatting instructions. In certain cases, they may contain some row-level data if it is included in filters.

Other data files, such as Excel or JSON files, are not encrypted with this feature unless they are converted to extracts before being published.

When extracts are downloaded from the server they are decrypted.

Performance Overview

Increase in Backgrounder Load

You may see a slight to moderate increase in backgrounder load when you turn on encryption at rest. Encryption and decryption are computationally intensive operations. Encryption at rest alters existing backgrounder jobs and introduces new jobs to run on backgrounder. The overall increase in backgrounder load depends on the number and size of affected extracts and how often the scenarios below apply.

- **Initial publishing**: When publishing workbooks or data sources using extracts that
should be encrypted, the encryption happens on the server’s backgrounders.

- **Extract refreshes from Tableau Server**: Full and incremental refreshes of encrypted extracts on Tableau Server will consume slightly more CPU.

- **Extract refreshes from Tableau Bridge and third-party applications (e.g., Informatica, Alteryx)**: These flows will require new encryption jobs, scheduled on the backgrounders for any refreshed extract, resulting in a slight to moderate increase in backgrounder load.

- **Encrypting and decrypting extracts in already published workbooks and data sources**: If the site setting for encryption at rest is set to Enable, users might choose to encrypt or decrypt extracts in already published workbooks and data sources on Tableau server. Depending on the number and size of extracts, this will add slight to moderate load on the backgrounders.

- **Changing a site’s encryption mode**: When switching a site’s setting for encryption at rest to Disable or Enforce, the backgrounder will, respectively, decrypt or encrypt all existing extracts on the site. Depending on the number and size of extracts, this may significantly increase the load on backgrounders until all extracts are unencrypted or encrypted.

- **Rotating encryption keys**: Rotating encryption keys results in the backgrounders re-encrypting all existing extracts published on that site, using fresh encryption keys. Depending on the number and size of extracts, this may significantly increase the load on backgrounders until all extracts are re-encrypted.

If running at or over capacity, consider:

- Adding additional backgrounder processes and resources.
- Letting users encrypt individual workbooks and data sources instead of enforcing encryption for the whole site or disable encryption at rest for sites where it isn’t necessary. Note that scheduled and ad hoc extract refreshes will take precedence over encryption and decryption jobs.

Increase in Viz Load Time and Worker Load

Query performance, for example, when loading or interacting with a viz or dashboard, will require the data being decrypted once, when loaded from disk to memory. This will result in
Tableau Server on Linux Administrator Guide

a slight increase in viz load time and CPU consumption on worker nodes for the first user loading a workbook. This will not affect other users accessing those workbooks at the same time because the data will already be decrypted in memory.

Impact on Backup and Restore

Encrypted extracts in backups remain encrypted. The size of backup files (.tbks) may increase up to 50-100% due to the ineffectiveness of compression on encrypted extracts. The size increase depends, among other factors, on the number of extracts that are encrypted. The time to restore a backup that contains encrypted extracts might increase slightly due to the time to exchange encryption keys.

If your Tableau Server installation has mostly or only encrypted extracts, consider disabling compression during backups to significantly improve the time backups take. To learn more about TSM backup, see tsm maintenance backup.

Enforce Encryption at Rest on a Site

Tableau Server administrators can enforce encryption of all extracts on their site.

1. In a web browser, sign in to Tableau Server as a server administrator.
2. Go to the site you want to configure.
3. Click Settings.
4. Scroll down to the Extract Encryption at Rest section.
   - Click Enforce to encrypt all extracts that are published and stored on the site. Encrypting all existing extracts stored on the site may take a while.
5. Click Save

Enable Encryption at Rest on a Site

Tableau Server administrators can allow users to specify to encrypt all extracts associated with particular published workbooks or data sources.

1. In a web browser, sign in to Tableau Server as a server administrator.
2. Go to the site you want to configure.
3. Click Settings.
4. Scroll down to the Extract Encryption at Rest section.
5. Click **Enable** to allow users to optionally encrypt extracts on the site.
   Changing to Enable will cancel pending decryption jobs and pending encryption jobs.
   No encryption jobs are created.
6. Click **Save**

**Disable Encryption at Rest on a Site**

1. In a web browser, sign in to Tableau Server as a server administrator.
2. Go to the site you want to configure.
3. Click **Settings**.
4. Scroll down to the Extract Encryption at Rest section.
5. Click **Disable** to not allow encrypted extracts on the site.
   Changing to Disable will decrypt all existing encrypted extracts. Decrypting all extracts
   stored on the site may take a while.
6. Click **Save**

**View Extract Encryption Mode for All Sites**

1. On a multi-site server, click **Manage all sites** on the site menu.
   
   **Note:** The **Manage all sites** option only displays when you are signed in as a server administrator.

2. Click **Sites**.
3. The encryption mode of each site is displayed in the **Extract encryption at rest** column.

**Encrypt or Decrypt Extracts for a Published Workbook or Data Source**

**Note:** The option to encrypt or decrypt the extracts associated with particular published
workbook or data source is only available when the site setting for encryption at rest is set to
**Enable**. When a site is set to **Disable**, all content is not encrypted. When a site is set to **Enforce**, all content is encrypted.

**Note:** You must be the owner or administrator.
Tableau Server on Linux Administrator Guide

Encrypt or Decrypt Multiple Items

1. Go to the Data Sources page.
2. Select the check box beside one or more data sources.
3. In the upper-left of the Data Sources page, click Actions.
4. Click Encrypt or Decrypt.

View Encryption Status for a Single Item

1. Sign in to the site.
2. Go to a single data source page.
3. Go to a single workbook page for a workbook containing embedded data sources.
4. The encryption status is displayed on the page.

Filter Data Sources by Encryption Status

1. In the site, click Explore.
2. At the top-right, click the Explore: Top-level Projects dropdown menu and select All Data Sources.
3. Click the filter icon.
4. Scroll down to the “Live or extract” section and select a filtering option: All, Live, Extracts, Unencrypted Extracts, Encrypted Extracts, Currently Encrypting, or Currently Decrypting.
5. Select the checkbox beside “Include .tde and .hyper files” if you want to include “Live to .tde file” and “Live to .hyper file” connections in your filter results.
Filter Workbooks by Encryption Status

1. In the site, click **Explore**.
2. At the top-right, click the **Explore: Top-level Projects** dropdown menu and select **All Workbooks**.
3. Click the filter icon.
4. Scroll down to the “Live or extract” section and select a filtering option: All, Live, Extracts, Published, Unencrypted Extracts, Encrypted Extracts, Currently Encrypting, or Currently Decrypting.
5. Select the checkbox beside “Include .tde and .hyper files” if you want to include “Live to .tde file” and “Live to .hyper file” connections in your filter results. Any workbooks that have at least one connection that matches the filter selection will be displayed.

View Status of Encrypt or Decrypt Extracts Background Tasks

1. In the site, click **Site Status**.
2. Click **Background Tasks for Non Extracts** to see completed and pending background task details.
   
   Note: **Background Tasks for Non Extracts** includes all tasks not related to extract refreshes, so it includes encryption jobs.
3. In the Task menu, select **Encrypt Extracts** or **Decrypt Extracts** and click **Apply**.
4. In the **Time Range** menu, select a range.
   
   You see "Encrypt Extracts" or "Decrypt Extracts" background tasks for all of your extract-based published data sources and workbooks.

The tabcmd Utility

The tabcmd command-line utility has commands and options to control extract encryption. For more information, see the tabcmd documentation.

Specify the extract encryption mode when you create a site

```
  tabcmd createsite <site-name> --extract-encryption-mode
  [enforced | enabled | disabled]
```
Specify the extract encryption mode when you edit a site

tabcmd editsite <site-name> --extract-encryption-mode [enforced | enabled | disabled]

Get the extract encryption mode when you list sites

tabcmd listsites --get-extract-encryption-mode

Encrypt extracts when you publish a workbook, data source, or extract to the server

tabcmd publish "filename.hyper" --encrypt-extracts

Decrypt all extracts on a site

**Note:** Depending on the number and size of extracts, this operation may consume significant server resources. Consider running this command outside of normal business hours.

```
tabcmd decryptextracts <site-name>
```

Encrypt all extracts on a site

**Note:** Depending on the number and size of extracts, this operation may consume significant server resources. Consider running this command outside of normal business hours.

```
tabcmd encryptextracts <site-name>
```

Reencrypt all extracts on a site with new encryption keys

If no site is specified, extracts on the default site will be reencrypted.
Note: Depending on the number and size of extracts, this operation may consume significant server resources. Consider running this command outside of normal business hours.

tabcmd reencryptextracts <site-name>

For more information, see reencryptextracts.

Tableau Server Rest API

With the Tableau Server REST API you can manage Tableau Server resources programmatically. You can use this access to create your own custom applications or to script interactions with Tableau Server resources.

To learn more, see Extract Encryption Methods.

Network Security

There are three main network interfaces in Tableau Server:

- **Client to Tableau Server**: The client can be a web browser, Tableau Mobile, Tableau Desktop, or the tabcmd utility.

- **Tableau Server to your database(s)**: To refresh data extracts or handle live database connections, Tableau Server needs to communicate with your database(s).

- **Server component communication**: This applies to distributed deployments only.

In most organization, Tableau Server is also configured to communicate with the internet and with a SMTP server.

Client to Tableau Server

A Tableau Server client can be a web browser, a device running Tableau Mobile, Tableau Desktop, or tabcmd commands. Communications between Tableau Server and its clients
use standard HTTP requests and responses. We recommend configuring Tableau Server for HTTPS for all communications. When Tableau Server is configured for SSL, all content and communications between clients are encrypted using SSL, and the HTTPS protocol is used for requests and responses.

By default, passwords are communicated from browsers and tabcmd to Tableau Server using 1024-bit public/private key encryption. This level of encryption is not considered robust enough for secure communications. Additionally, this method, where a public key is sent to the recipient in the clear and without network layer authentication is susceptible to man-in-the-middle attacks.

To adequately secure network traffic from clients to Tableau Server, you must configure SSL with a certificate from a trusted certificate authority.

See Configure SSL for External HTTP Traffic to and from Tableau Server.

Client access from the Internet

We recommend a gateway proxy server to enable secure client access from the internet to your Tableau Server. We do not recommend running Tableau Server in a DMZ or otherwise outside your protected, internal network.

Configure a reverse proxy server, with SSL enabled, to handle all inbound traffic from the internet. In this scenario, the reverse proxy is the only external IP address (or range of addresses if multiple reverse proxies are load-balancing inbound requests) that Tableau Server will communicate with. Reverse proxies are transparent to requesting clients, thereby obfuscating Tableau Server network information and simplifying client configuration.

For configuration information, see Configuring Proxies for Tableau Server.

Clickjack Protection

By default, Tableau Server has clickjack protection enabled. This helps prevent certain types of attacks in which the attacker overlays a transparent version of a page on top of an innocuous-looking page in order to lure a user into clicking links or entering information. With click-
jack protection enabled, Tableau Server imposes certain restrictions on embedding views. For more information, see Clickjack Protection.

Tableau Server to your database

Tableau Server makes dynamic connections to databases to process result sets and refresh extracts. It uses native drivers to connect to databases whenever possible and relies on a generic ODBC adapter when native drivers are unavailable. All communications to the database are routed through these drivers. As such, configuring the driver to communicate on non-standard ports or provide transport encryption is part of the native driver installation. This type of configuration is transparent to Tableau.

When a user stores credentials for external data sources on Tableau Server, they are stored encrypted in Tableau Server’s internal database. When a process uses those credentials to query the external data source, the process retrieves the encrypted credentials from the internal database and decrypts them in process.

Tableau Server to the internet

In some cases, where users connect to external data sources, such as the Tableau map servers, then Tableau Server will need to connect to the internet. We recommend that you run all components of Tableau inside your protected network. Therefore, connections to the internet may require that you configure Tableau Server to use a forward proxy.

Tableau Server to a SMTP server

You can configure Tableau Server to send event notifications to administrators and users. As of version 2019.4, Tableau Server supports TLS for the SMTP connection. See Configure SMTP Setup.

Communication with the repository

You can configure Tableau Server to use Secure Sockets Layer (SSL) for encrypted communications on all traffic that is exchange with the Postgres repository to and from other
By default, SSL is disabled for communications between server components and the repository.

For more information, see tsm security repository-ssl enable

Server component communication in a cluster

There are two aspects to communication between Tableau Server components in a distributed server installation: trust and transmission. Each server in a Tableau cluster uses a stringent trust model to ensure that it is receiving valid requests from other servers in the cluster. Computers in the cluster running a gateway process accept requests from third parties (clients), unless they are fronted by a load balancer, in which case the load balancer receives the requests. Servers not running a gateway process only accept requests from other trusted members of the cluster. Trust is established by a whitelist of IP address, port, and protocol. If any of these are invalid, the request is ignored. All members of the cluster can communicate with each other.

When a user stores credentials for external data sources on Tableau Server, they are stored encrypted in Tableau Server’s internal database. When a process uses those credentials to query the external data source, the process retrieves the encrypted credentials from the internal database and decrypts them in process.

Clickjack Protection

Tableau Server includes protection against clickjack attacks. Clickjacking is a type of attack against web pages in which the attacker tries to lure users into clicking or entering content by displaying the page to attack in a transparent layer over an unrelated page. In the context of Tableau Server, an attacker might try to use a clickjack attack to capture user credentials or to get an authenticated user to change settings on your server. For more information about clickjack attacks, see Clickjacking on the Open Web Application Security Project website.
**Note:** Clickjack protection was available in previous versions of Tableau Server, but was disabled by default. New installations of Tableau Server 9.1 and later will always have clickjack protection on unless you explicitly disable it.

Effects of clickjack protection

When clickjack protection is enabled on Tableau Server, the behavior of pages loaded from Tableau Server changes in the following ways:

- **Tableau Server adds the** `X-Frame-Options: SAMEORIGIN` **header to certain responses from the server.** In the current versions of most browsers, this header prevents the content from being loaded into an `<iframe>` element, which helps prevent clickjacking attacks.

- **The top-level page from Tableau Server cannot be loaded in `<iframe>` elements.** This includes the sign-in page. One consequence is that you cannot host Tableau Server pages in an application that you create.

- Only views can be embedded.

- If an embedded view requires data source credentials, a message is displayed in the `<iframe>` element with a link to open the view in a secure window where the user can safely enter credentials. Users should always verify the address of the opened window before entering credentials.

- Views can be loaded only if they include the `:embed=y` parameter in the query string, as in this example:

  
  http://<server>/views/Sales/CommissionModel?:embed=y

  
**Note:** When clickjack protection is enabled, embedded views that use the URL copied from the browser address bar might not load. These view URLs usually
Disabling clickjack protection

You should leave clickjack protection enabled unless it is affecting how your users work with Tableau Server. If you want to disable clickjack protection, use the following `tsm` commands:

1. `tsm configuration set -k wgserver.clickjack_defense.enabled -v false`
2. `tsm pending-changes apply`

   The `pending-changes apply` command displays a prompt to let you know this will restart Tableau Server if the server is running. The prompt displays even if the server is stopped, but in that case there is no restart. You can suppress the prompt using the `--ignore-prompt` option, but this does not change the restart behavior. For more information, see `tsm pending-changes apply`.

HTTP Response Headers

Tableau Server supports some of the response headers specified in the OWASP Secure Headers Project.

This topic describes how to configure the following response headers for Tableau Server:

- HTTP Strict Transport Security (HSTS)
- Referrer-Policy
- X-Content-Type-Options
- X-XSS-Protection

Tableau Server also supports the Content Security Policy (CSP) standard. CSP configuration is not covered in this topic. See Content Security Policy.
Configuring response headers

All response headers are configured with the tsm configuration set command.

When you are finished configuring response headers, run tsm pending-changes apply.

The `pending-changes apply` command displays a prompt to let you know this will restart Tableau Server if the server is running. The prompt displays even if the server is stopped, but in that case there is no restart. You can suppress the prompt using the `--ignore-prompt` option, but this does not change the restart behavior. For more information, see `tsm pending-changes apply`.

HTTP Strict Transport Security (HSTS)

HSTS forces clients connecting to Tableau Server to connect with HTTPS. For more information see the OWASP entry, [HTTP Strict Transport Security (HSTS)](https://owasp.org/www-community/standards/Model/BestPractices/05-HSTS).

Options

gateway.http.hsts

Default value: `false`

The HTTP Strict Transport Security (HSTS) header forces browsers to use HTTPS on the domain where it is enabled.

gateway.http.hsts_options

Default value: `"max-age=31536000"`

By default, HSTS policy is set for one year (31536000 seconds). This time period specifies the amount of time in which the browser will access the server over HTTPS.

Referrer-Policy

Beginning in 2019.2, Tableau Server includes the ability to configure Referrer-Policy HTTP header behavior. This policy is enabled with a default behavior that will include the origin URL for all "secure as" connections (policy `no-referrer-when-downgrade`). In
Tableau Server on Linux Administrator Guide

previous versions, the Referrer-Policy header was not included in responses sent by Tableau Server. For more information about the various policy options that Referrer-Policy supports, see the OWASP entry, Referrer-Policy.

Options

gateway.http.referrer_policy_enabled

Default value: true

To exclude the Referrer-Policy header from responses sent by Tableau Server, set this value to false.

gateway.http.referrer_policy

Default value: no-referrer-when-downgrade

This option defines the referrer policy for Tableau Server. You may specify any of the policy value strings listed in the Referrer-Policy table on the OWASP page.

X-Content-Type-Options

The X-Content-Type-Options response HTTP header specifies that the MIME type in the Content-Type header should not be changed by the browser. In some cases, where MIME type is not specified, a browser may attempt to determine the MIME type by evaluating the characteristics of the payload. The browser will then display the content accordingly. This process is referred to as "sniffing." Misinterpreting the MIME type can lead to security vulnerabilities.

For more information see the OWASP entry, X-Content-Type-Options.

Option

gateway.http.x_content_type_nosniff

Default value: true
The X-Content-Type-Options HTTP header is set to 'nosniff' by default with this option.

X-XSS-Protection

The HTTP X-XSS-Protection response header is sent to the browser to enable cross-site scripting (XSS) protection. The X-XSS-Protection response header overrides configurations in cases where users have disabled XXS protection in the browser.

For more information see the OWASP entry, X-XSS-Protection.

Option

gateway.http.x_xss_protection

  Default value: true

  The X-XSS-Protection response header is enabled by default with this option.

Content Security Policy

Tableau Server supports the Content Security Policy (CSP) standard. CSP is intended to be an additional layer of security against cross-site scripting and other malicious web-based attacks. CSP is implemented as a HTTP response header that allows you to specify where external resources, such as scripts and images, can be safely loaded from.

See the Mozilla website for more information about CSP.

Note: If your Tableau Server is configured to use SAML for user authentication, then do not enable CSP. Running CSP on Tableau Server with SAML is not supported.

Configure and enable CSP

CSP is configured and enabled using the tsm configuration set Options command. If you are running Tableau Server in a distributed deployment, run these commands on the initial
Step 1: Set default directives

Tableau Server includes the set of default directives in the table below.

To set a directive, use the following tsm syntax:

```bash
tsm configuration set -k content_security_policy.-directive.<directive_name> -v "<value>"
```

For example, to set the `connect_src` directive, run the following command:

```bash
tsm configuration set -k content_security_policy.-directive.connect_src -v "* unsafe-inline"
```

<table>
<thead>
<tr>
<th>Option</th>
<th>Default value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>content_security_policy.-directive.default_src</td>
<td>'none'</td>
<td>Serves as a fallback for the other fetch directives. Valid values for <code>default_src</code>.</td>
</tr>
<tr>
<td>content_security_policy.-directive.connect_src</td>
<td>*</td>
<td>Restricts the URLs which can be loaded using script interfaces. Valid values for <code>connect_src</code>.</td>
</tr>
<tr>
<td>content_security_policy.directive.script_src</td>
<td>*</td>
<td>Specifies valid sources for JavaScript. Valid values for <code>script_src</code>.</td>
</tr>
<tr>
<td>content_security_policy.directive.style_src</td>
<td>* 'unsafe-inline'</td>
<td>Specifies valid sources for stylesheets. Valid values for style_src.</td>
</tr>
<tr>
<td>content_security_policy.directive.img_src</td>
<td>* data:</td>
<td>Specifies valid sources of images and favicons. Valid values for img_src.</td>
</tr>
<tr>
<td>content_security_policy.directive.font_src</td>
<td>* data:</td>
<td>Specifies valid sources for fonts loaded using @font-face. Valid values for font_src.</td>
</tr>
<tr>
<td>content_security_policy.-directive.frame_src</td>
<td>* data:</td>
<td>Specifies valid sources for nested browsing contexts loading using elements such as &lt;frame&gt; and &lt;iframe&gt;. Valid values for frame_src.</td>
</tr>
<tr>
<td>content_security_policy.-directive.object_src</td>
<td>data:</td>
<td>Specifies valid sources for the &lt;object&gt;, &lt;embed&gt;, and &lt;applet&gt; elements. Valid values for object_src.</td>
</tr>
<tr>
<td>content_security_policy.-</td>
<td>/vizql/csp-report</td>
<td>Instructs the user agent to</td>
</tr>
</tbody>
</table>
Report directive.report_uri attempts to violate the CSP. These violation reports consist of JSON documents sent via an HTTP POST request to the specified URI.

Valid values for report_uri.

Step 2: Add additional directives (optional)

The default directives included with Tableau Server are a subset of directives that are supported by CSP.

For a full list of supported CSP directives, go to https://developer.mozilla.org/en-US/docs/Web/HTTP/Headers/Content-Security-Policy.

You can add directives to the existing default set, by using adding the new directive in the content_security_policy.directive namespace. You must include the --force-keys parameter when adding new directives. The syntax is as follows:

```bash
tsm configuration set -k content_security_policy.directive.<new_directive_name> -v "<value>" --force-keys
```

For example, to add the worker-src directive, run the following command:

```bash
tsm configuration set -k content_security_policy.directive.worker-src -v "none" --force-keys
```

Step 3: Specify report-only directives (optional)

You can configure CPS to report some directives and to enforce others. When you set content_security_policy.enforce_enabled to true, then all directives are enforced (even if
content_security_policy.report_only_enable is also set to true).

To specify directives as "report-only" and not enforced, add the directives to the report_only_directive namespace. You must include the --force-keys parameter when adding new directives. The syntax is as follows:

```bash
tsm configuration set -k content_security_policy.report_only_directive.<directive_name> -v "<value>" --force-keys
```

For example, to report only on the script_src directive, run the following command:

```bash
tsm configuration set -k content_security_policy.report_only_directive.script_src -v " http://*.example.com" --force-keys
```

**Step 4: Enable CSP on Tableau Server**

After you have configured directives, enable CSP on Tableau Server.

The following options are used to enable enforcement or report only mode for the directives you have set.

<table>
<thead>
<tr>
<th>Option</th>
<th>Default value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>content_security_policy.enforce_enabled</td>
<td>false</td>
<td>Adds a CSP header to all requests so that any violation will be enforced by the browser.</td>
</tr>
<tr>
<td>content_security_policy.report_only_enabled</td>
<td>true</td>
<td>Adds a CSP header to all requests so that any violation will be recorded in our vizql-client logs, but will not be enforced by the browser.</td>
</tr>
</tbody>
</table>
To enable enforcement of the CSP directives that you’ve specified, run the following command:

```
> tsm configuration set -k content_security_policy.enforce_enabled -v true
```

**Step 5: Run tsm pending-changes apply**

When you are finished configuring CSP, run `tsm pending-changes apply`.

The `pending-changes apply` command displays a prompt to let you know this will restart Tableau Server if the server is running. The prompt displays even if the server is stopped, but in that case there is no restart. You can suppress the prompt using the `--ignore-prompt` option, but this does not change the restart behavior. For more information, see `tsm pending-changes apply`.

**View CSP report**

To view CSP violations for a given viz, load the viz in a browser that includes developer tools. This example uses the Chrome browser.

1. Load a test viz with violations that is hosted on the Tableau Server deployment where you configured CSP.
2. Enter `CTRL+Shift+I` to open the developer tools in Chrome.
3. Click the **Network** tab.
4. In the **Filter** field, enter `csp-report`, and then click **Find All**.
   - If there are no violations then the search will not return any CSP reports.
   - If there are violations, click the Headers tab in the results pane and scroll to the bottom to view Request Payload.
SSL

SSL (Secure Sockets Layer) is a standard security technology that establishes an encrypted link between a web server and clients. To use SSL, you need to install an SSL certificate on Tableau Server.

You can configure Tableau Server to use SSL in the following ways:

- Use SSL for external HTTP traffic.
- Use mutual (two-way) SSL between clients (Tableau Desktop, web browsers, and tabcmd.exe) and Tableau Server.
- Use SSL for all HTTP traffic between internal server components and the repository.

If you are using mutual SSL, each client also needs a certificate.

**Note:** Tableau Server uses SSL for user authentication. SSL is not used to handle permissions and authorization to content (data sources and workbooks) hosted on Tableau Server.

Configure SSL for External HTTP Traffic to and from Tableau Server

You can configure Tableau Server to use Secure Sockets Layer (SSL) encrypted communications on all external HTTP traffic. Setting up SSL ensures that access to Tableau Server is secure and that sensitive information passed between the server and Tableau clients—such as Tableau Desktop, the REST API, and so on—is protected. Steps on how to configure the server for SSL are described this topic; however, you must first acquire a certificate from a trusted authority, and then import the certificate files into Tableau Server.

For a basic primer about SSL and Tableau Server, see Using SSL to encrypt Tableau Server communication in the Everybody’s Install Guide.

Mutual SSL authentication is not supported on Tableau Mobile.
Acquire an Apache SSL certificate from a trusted authority (for example, Verisign, Thawte, Comodo, GoDaddy). You can also use an internal certificate issued by your company. Wild-card certificates, which allow you to use SSL with many host names within the same domain, are also supported.

When you acquire an SSL certificate for external communication to and from Tableau Server, follow these guidelines and requirements:

- All certificate files must be valid PEM-encoded X509 certificates with the extension .crt.

- Use a SHA-2 (256 or 512 bit) SSL certificate. Most browsers no longer connect to a server that presents an SHA-1 certificate.

- In addition to the certificate file, you must also acquire a corresponding SSL certificate key file. The key file must be a valid RSA or DSA private key file (with the extension .key by convention).

  You can choose to passphrase-protect the key file. The passphrase you enter during configuration will be encrypted while at rest. However, if you want to use the same certificate for SSL and SAML, you must use a key file that is not passphrase protected.

- SSL certificate chain file: A certificate chain file is required for Tableau Desktop on the Mac and for Tableau Prep Builder on the Mac and Tableau Prep Builder on Windows. The chain file is also required for the Tableau Mobile app if the certificate chain for Tableau Server is not trusted by the iOS or Android operating system on the mobile device. The chain file is a concatenation of all of the certificates that form the certificate chain for the server certificate. All certificates in the file must be x509 PEM-encoded and the file must have a .crt extension (not .pem).

- For multiple sub-domains, Tableau Server supports wildcard certificates.
Tableau Server supports certificates that list multiple domains, IP addresses, or host names in the Subject Alternative Names (SAN) field.

Note: If you plan to configure Tableau Server for single-sign on using SAML, see Using SSL certificate and key files for SAML in the SAML requirements to help determine whether to use the same certificate files for both SSL and SAML.

Configuring SSL for a Cluster

You can configure a Tableau Server cluster to use SSL. If the initial node is the only one running the gateway process (which it does by default), you need to configure SSL only on that node, using the steps described in this topic.

SSL with multiple gateways

A highly available Tableau Server cluster can include multiple gateways, fronted by a load balancer. If you are configuring this type of cluster for SSL, you have the following choices:

- **Configure the load balancer for SSL**: Traffic is encrypted from the client web browsers to the load balancer. Traffic from the load balancer to the Tableau Server gateway processes is not encrypted. No SSL configuration in Tableau Server is required by you. It’s all handled by the load balancer.

- **Configure Tableau Server for SSL**: Traffic is encrypted from the client web browsers to the load balancer, and from the load balancer to the Tableau Server gateway processes. For more information, continue to the following section.

Additional configuration information for Tableau Server cluster environments

When you want to use SSL on all Tableau Server nodes that run a gateway process, you complete the following steps.
1. Configure the external load balancer for SSL passthrough.

Or if you want to use a port other than 443, you can configure the external load balancer to terminate the non-standard port from the client. In this scenario, you would then configure the load balancer to connect to Tableau Server over port 443. For assistance, refer to the documentation provided for the load balancer.

2. Make sure the SSL certificate is issued for the load balancer’s host name.

3. Configure the initial Tableau Server node for SSL.

4. If you are using mutual SSL, upload the SSL CA certificate file. See `tsm authentication mutual-ssl <commands>`.

SSL certificate and key files will be distributed to each node as part of the configuration process.

Prepare the environment

When you get the certificate files from the CA, save them to a location accessible by Tableau Server, and note the names of the certificate .crt and .key files and the location where you save them. You’ll need to provide this information to Tableau Server when you enable SSL.

A common practice is to place a copy of the certificate files in a location that’s within the Tableau Server directory tree. For example:

`/var/opt/tableau/tableau_server/data/ssl`

Name the directory whatever is appropriate depending on whether you’ll use the certificate files only for SSL or also for configuring SAML authentication.

**Note:** If you think you might want to use the same certificate for SSL and SAML, see the Certificate and identity provider (IdP) requirements in the SAML Requirements topic.
Configure SSL on Tableau Server

Use the method you’re most comfortable with.

Use the TSM web interface

1. Open TSM in a browser:

   https://<tsm-computer-name>:8850. For more information, see Sign in to Tableau Services Manager Web UI.

2. On the Configuration tab, select Security > External SSL.

   Note: If you are updating or changing an existing configuration, click Reset to clear the existing settings before proceeding.

3. Under External web server SSL, select Enable SSL for server communication.

4. Upload the certificate and key files, and if required for your environment, upload the chain file and enter the passphrase key:

   If you are running Tableau Server in a distributed deployment, then these files will be automatically distributed to each appropriate node in the cluster.

5. Click Save Pending Changes.
6. Click **Pending Changes** at the top of the page:

7. Click **Apply Changes and Restart**.

Use the TSM CLI

After you have copied the certificate files to the local computer, run the following commands:

```
 tsm security external-ssl enable --cert-file <path-to-file.crt> 
 --key-file <path-to-file.key>
 tsm pending-changes apply
```

See the command reference at tsm security external-ssl enable to determine whether you want to include additional options for external-ssl enable. Tableau has specific recommendations for the `--protocols` option.

The `external-ssl enable` command imports the information from the .crt and .key files. If you run this command on a node in a Tableau Server cluster, it also distributes the information to any other gateway node.

The `pending-changes apply` command displays a prompt to let you know this will restart Tableau Server if the server is running. The prompt displays even if the server is stopped, but in that case there is no restart. You can suppress the prompt using the `--ignore-prompt` option, but this does not change the restart behavior. For more information, see tsm pending-changes apply.

Port redirection and logging

After the server has been configured for SSL, it accepts requests to the non-SSL port (default is port 80) and automatically redirects to the SSL port 443.
**Tableau Server on Linux Administrator Guide**

**Note:** Tableau Server supports only port 443 as the secure port. It cannot run on a computer where another application is using port 443.

SSL errors are logged in the at the following location. Use this log to troubleshoot validation and encryption issues:

/var/opt/tableau/tableau_server-
data/tabsvc/logs/httpd/error.log

**Add SSL port to the local firewall**

If you are running a local firewall, you must add the SSL port to the firewall on Tableau Server. The example below describes how to configure the firewall running on RHEL/CentOS distributions. The example uses **Firewalld**, which is the default firewall on CentOS.

1. **Start firewalld**:
   
   ```bash
   sudo systemctl start firewalld
   ```

2. **Add port 443 for SSL**:
   
   ```bash
   sudo firewall-cmd --permanent --add-port=443/tcp
   ```

3. **Reload the firewall and verify the settings**:
   
   ```bash
   sudo firewall-cmd --reload
   
   sudo firewall-cmd --list-all
   ```

**Change or update SSL certificate**

After you have configured SSL, you may need to periodically update the certificate. In some cases, you may need change the certificate for operational changes in your IT environment.
In either case, you must use TSM to replace the SSL certificate that has already been configured for external SSL.

Do not copy a new certificate to the file directory on the operating system. Rather, when you add the certificate with either the TSM web UI or the `tsm security external-ssl enable` command, the certificate file is copied to the appropriate certificate store. In a distributed deployment, the certificate is also copied across the nodes in the cluster.

To change or update the SSL certificate (and the corresponding key file if required), follow the steps in the previous section of this topic, Configure SSL on Tableau Server.

Example: SSL Certificate - Generate a Key and CSR

**Important:** This example is intended to provide general guidance to IT professionals who are experienced with SSL requirements and configuration. The procedure described in this article is just one of many available methods you can use to generate the required files. The process described here should be treated as an example and not as a recommendation.

When you configure Tableau Server to use Secure Sockets Layer (SSL) encryption, this helps ensure that access to the server is secure and that data sent between Tableau Server and Tableau Desktop is protected.

Looking for Tableau Server on Windows? See Example: SSL Certificate - Generate a Key and CSR.

Tableau Server uses Apache, which includes OpenSSL. You can use the OpenSSL toolkit to generate a key file and Certificate Signing Request (CSR) which can then be used to obtain a signed SSL certificate.

Steps to generate a key and CSR

To configure Tableau Server to use SSL, you must have an SSL certificate. To obtain the SSL certificate, complete the steps:

1. Generate a key file.
2. Create a Certificate Signing Request (CSR).
3. Send the CSR to a certificate authority (CA) to obtain an SSL certificate.
4. Use the key and certificate to configure Tableau Server to use SSL.

You can find additional information on the SSL FAQ page on the Apache Software Foundation website.

Configure a certificate for multiple domain names

Tableau Server allows SSL for multiple domains. To set up this environment, you need to modify the OpenSSL configuration file, openssl.conf, and configure a Subject Alternative Name (SAN) certificate on Tableau Server. See For SAN certificates: modify the OpenSSL configuration file below.

Generate a key

Generate a key file that you will use to generate a certificate signing request.

1. Change to the Apache directory for Tableau Server:

   /opt/tableau/tableau_server/packages/apache.<version>

2. Run the following command to create the key file:

   openssl genrsa -out <yourcertname>.key 4096

   **Note:** This command uses a 4096-bit length for the key. You should choose a bit length that is at least 2048 bits because communication encrypted with a shorter bit length is less secure. If a value is not provided, 512 bits is used.

Create a certificate signing request to send to a certificate authority

Use the key file you created in the procedure above to generate the certificate signing request (CSR). You send the CSR to a certificate authority (CA) to obtain a signed
Important: If you want to configure a SAN certificate to use SSL for multiple domains, first complete the steps in For SAN certificates: modify the OpenSSL configuration file below, and then return to here to generate a CSR.

1. Run the following command to create a certificate signing request (CSR) file:

   ```bash
   openssl req -new -key yourcertname.key -out yourcertname.csr -config ../conf/openssl.cnf
   ```

2. When prompted, enter the required information.

   Note: For Common Name, type the Tableau Server name. The Tableau Server name is the URL that will be used to reach the Tableau Server. For example, if you reach Tableau Server by typing `tableau.example.com` in the address bar of your browser, then `tableau.example.com` is the common name. If the common name does not resolve to the server name, errors will occur when a browser or Tableau Desktop tries to connect to Tableau Server.

Send the CSR to a certificate authority to obtain an SSL certificate

Send the CSR to a commercial certificate authority (CA) to request the digital certificate. For information, see the Wikipedia article Certificate authority and any related articles that help you decide which CA to use.

Use the key and certificate to configure Tableau Server

When you have both the key and the certificate from the CA, you can configure Tableau Server to use SSL. For the steps, see Configure External SSL.
For SAN certificates: modify the OpenSSL configuration file

In a standard installation of OpenSSL, some features are not enabled by default. To use SSL with multiple domain names, before you generate the CSR, complete these steps to modify the openssl.cnf file.

1. Navigate to the Apache conf folder for Tableau Server.
   
   For example: /opt/tableau/tableau_server-packages/apache.<version_code>/conf

2. Open openssl.cnf in a text editor, and find the following line: req_extensions = v3_req

   This line might be commented out with a hash sign (#) at the beginning of the line.

   ```
   req_extensions = v3_req
   #req_extensions = v3_req # The extensions to add to a certificate request
   [ req_distinguished_name ]
   countryName = Country Name (2 letter code)
   ```

   If the line is commented out, uncomment it by removing the # and space characters from the beginning of the line.

3. Move to the [ v3_req ] section of the file. The first few lines contain the following text:

   ```
   # Extensions to add to a certificate request
   basicConstraints = CA:FALSE
   keyUsage = nonRepudiation, digitalSignature, keyEncipherment
   ```

   After the keyUsage line, insert the following line:

   ```
   subjectAltName = @alt_names
   ```
If you’re creating a self-signed SAN certificate, do the following to give the certificate permission to sign the certificate:

a. Add the `cRLSign` and `keyCertSign` to the `keyUsage` line so it looks like the following:
   ```
   keyUsage = nonRepudiation, digitalSignature, 
   keyEncipherment, cRLSign, keyCertSign
   ```

b. After the `keyUsage` line, add the following line:
   ```
   subjectAltName = @alt_names
   ```

4. In the `[alt_names]` section, provide the domain names you want to use with SSL.
   ```
   DNS.1 = [domain1]
   DNS.2 = [domain2]
   DNS.3 = [etc]
   ```

   The following image shows the results highlighted, with placeholder text that you would replace with your domain names.

   ![Highlighted code]

5. Save and close the file.

6. Complete the steps in Create a certificate signing request to send to a certificate authority section, above.
Configure SSL for Internal Postgres Communication

You can configure Tableau Server to use SSL (TLS) for encrypted communication between the Postgres repository and other server components. By default, communication that is internal to Tableau Server components is not encrypted.

While you enable support for internal SSL, you can also configure support for direct connections to the repository from Tableau clients, such as Tableau Desktop, Tableau Mobile, REST API, web browsers.

Use the TSM web interface

1. As a server administrator, open TSM in a browser:

   https://<tsm-computer-name>:8850

   For more information, see Sign in to Tableau Services Manager Web UI.

2. On the Configuration tab, select Security > Repository SSL.

   ![Repository SSL Configuration](image)

3. Select one of the options for using repository SSL.

   - **Required for all connections**—uses SSL for internal Tableau Server communication, and requires SSL for Tableau clients that connect directly to the
Tableau Server on Linux Administrator Guide

repository, including those using the **tableau** or **readonly** user.

**Important:** If you select this option, you must also complete the steps in Configure Postgres SSL to Allow Direct Connections from Clients, to place the certificate files in the correct location on the client computers.

- **Optional for user connections**—uses SSL for internal Tableau Server communication, and supports but does not require SSL for direct connections to the server from Tableau clients.

- **Off for all connections (default)**—Internal server communication is not encrypted, and SSL is not required for direct connections from clients.

4. Click **OK**.

The first two options generate the server’s certificate files, `server.crt` and `server.key`, and place them in the following location.

`/var/opt/tableau/tableau_server/data/tabsvc/config/pgsql_<version>/security`

Use this .crt file if you need to configure clients for direct connections.

**Use the TSM CLI**

To enable SSL for internal traffic among the server components, run the following commands:

```
    tsm security repository-ssl enable
```

```
    tsm pending-changes apply
```
What the command does

`repository-ssl enable` generates the server's certificate files, which it places in the following location:

```
/var/opt/tableau/tableau_server/data/tabsvc/config/pgsql_<version>/security
```

By default, this command sets Tableau Server to require SSL for traffic between the repository and other server components, as well as for direct connections from Tableau clients (including for connections through the `tableau` or `readonly` users).

To complete the configuration, you must also do the steps described in Configure Postgres SSL to Allow Direct Connections from Clients, to place the certificate files in the correct location on the client computers.

The `pending-changes apply` command displays a prompt to let you know this will restart Tableau Server if the server is running. The prompt displays even if the server is stopped, but in that case there is no restart. You can suppress the prompt using the `--ignore-prompt` option, but this does not change the restart behavior. For more information, see `tsm pending-changes apply`.

Option for repository-ssl enable

If you want to require SSL only for internal Tableau Server communication, and not for direct connections from client apps, use the following option with the `repository-ssl enable` command:

`--internal-only`

Cluster environments

If you run `repository-ssl enable` on a node in a cluster, it copies the required certificate file to the same location on each other node.
Tableau Server on Linux Administrator Guide

For more information about downloading the public certificate for direct connections, see Configure Postgres SSL to Allow Direct Connections from Clients.

Configure Postgres SSL to Allow Direct Connections from Clients

When Tableau Server is configured to use SSL for internal communication with the Postgres repository, you can also require SSL for Tableau clients that connect directly to the repository. Direct connections include those using the `tableau` user or the `readonly` user. Examples of Tableau clients include Tableau Desktop, Tableau Mobile, REST API, web browsers.

1. Run the following commands:

   tsm security repository-ssl enable
   
   tsm pending-changes apply

   This enables internal SSL support, generates new server certificate and key files, and requires all Tableau clients to use SSL to connect to the repository. For additional repository-ssl commands and options, see tsm security.

   The `pending-changes apply` command displays a prompt to let you know this will restart Tableau Server if the server is running. The prompt displays even if the server is stopped, but in that case there is no restart. You can suppress the prompt using the `--ignore-prompt` option, but this does not change the restart behavior. For more information, see tsm pending-changes apply.

2. (Optional) If you have configured your client computer to validate Postgres SSL connections, then you must import the certificate that is generated by Tableau Server onto the computers running Tableau Desktop. For each client computer that will connect directly to the repository, do the following:

   - Copy the `server.crt` file to the client computer. You can find this file in the following directory:
Configure Mutual SSL Authentication

Using mutual SSL, you can provide users of Tableau Desktop and other approved Tableau clients a secure, direct-access experience to Tableau Server. With mutual SSL, when a client with a valid SSL certificate connects to Tableau Server, Tableau Server confirms the existence of the client certificate and authenticates the user, based on the user name in the client certificate. If the client does not have a valid SSL certificate, Tableau Server can refuse the connection. You can also configure Tableau Server to fall back to username/password authentication if mutual SSL fails.

Mutual SSL authentication is not supported on Tableau Mobile.

Use the TSM web interface

1. Configure SSL for External HTTP Traffic to and from Tableau Server.

2. Open TSM in a browser:

   https://<tsm-computer-name>:8850. For more information, see Sign in to Tableau Services Manager Web UI.


4. Under Authentication Method, select Mutual SSL in the drop-down menu.
5. Under Mutual SSL, select **Use mutual SSL and automatic sign in with client certificates**.

6. Click **Select File** and upload your certificate authority (CA) issued certificate to the server.

   This certificate must be a valid PEM-encoded x509 certificate with the extension .crt.

7. Enter remaining SSL configuration information for your organization.

   **Username format:** When Tableau Server is configured for mutual SSL, the server gets the user name from the client certificate, so it can establish a direct sign-in for the client user. The name that Tableau Server uses depends on how Tableau Server is configured for user authentication:

   - Local Authentication—Tableau Server uses the UPN (User Principal Name) from the certificate.
   - Active Directory (AD)—Tableau Server uses LDAP (Lightweight Directory Access Protocol) to get the user name.

   Alternatively, you can set Tableau Server to use the CN (Common Name) from the client certificate.

8. Click **Save Pending Changes** after you've entered your configuration information.
9. Click **Pending Changes** at the top of the page:

![Pending Changes](image)

10. Click **Apply Changes and Restart**.

Use the TSM CLI

**Step 1: Require SSL for external server communication**

To configure Tableau Server to use SSL for external communication between Tableau Server and web clients, run the `external-ssl enable` command as follows, providing the names for the server certificate’s .crt and .key files:

```
tsm security external-ssl enable --cert-file <file.crt> --key-file <file.key>
```

- For `--cert-file` and `--key-file`, specify the location and file name where you saved the server’s CA-issued SSL certificate (.crt) and key (.key) files.

- The above command assumes the you are signed in as a user that has the **Server Administrator** site role on Tableau Server. You can instead use the `-u` and `-p` parameters to specify an administrator user and password.

- If the certificate key file requires a passphrase, include the `--passphrase` parameter and value.

**Step 2: Use mutual SSL**

Add mutual authentication between the server and each client, and allow for Tableau client users to be authenticated directly after the first time they provide their credentials.
1. Run the following command:

   `tsm authentication mutual-ssl configure -cert-file <file.crt>`

   For `-cert-file`, specify the location and file name of the server’s CA certificate (.crt) file, as in the previous step for external SSL.

   See the remaining sections in this article for any additional options you might want to include with the `mutual-ssl configure` command.

2. Run the following commands to enable mutual SSL and apply the changes:

   `tsm authentication mutual-ssl enable`

   `tsm pending-changes apply`

   The `pending-changes apply` command displays a prompt to let you know this will restart Tableau Server if the server is running. The prompt displays even if the server is stopped, but in that case there is no restart. You can suppress the prompt using the `--ignore-prompt` option, but this does not change the restart behavior. For more information, see `tsm pending-changes apply`.

**Additional options for mutual SSL**

You can use `mutual-ssl configure` to configure Tableau Server to support the following options.

For more information, see `tsm authentication mutual-ssl <commands>`.

**Fallback authentication**

When Tableau Server is configured for mutual SSL, authentication is automatic and clients must have a valid certificate. You can configure Tableau Server to allow a fallback option, to accept user name and password authentication.

   `tsm authentication mutual-ssl configure -fb true`
Tableau Server accepts username and password authentication from REST API clients, even if the above option is set to `false`.

User name mapping

When Tableau Server is configured for mutual SSL, the server authenticates the user directly by getting the user name from their client certificate. The name that Tableau Server uses depends on how the server is configured for user authentication:

- **Local Authentication**—uses the UPN (User Principal Name) from the certificate.
- **Active Directory (AD)**—uses LDAP (Lightweight Directory Access Protocol) to get the user name.

You can override either of these defaults to set Tableau Server to use the common name.

```
tsm authentication mutual-ssl configure -m cn
```

For more information, see Mapping a Client Certificate to a User During Mutual Authentication

Certificate Revocation List (CRL)

You might need to specify a CRL if you suspect that a private key has been compromised, or if a certificate authority (CA) did not issue a certificate properly.

```
tsm authentication mutual-ssl configure -rf <revoke-file.pem>
```

Mapping a Client Certificate to a User During Mutual Authentication

When you use mutual (two-way) SSL authentication, the client presents its certificate to Tableau Server as part of the authentication process. Tableau Server then maps user information in the client certificate to a known user identity. The strategy that Tableau Server uses to perform client mapping depends on the content of your organization’s client certificates.
This topic discusses the ways information in a client certificate can map to a user identity and how to change the way Tableau Server performs that mapping. To understand how the mapping happens and whether you need to change it, you must know how client certificates are structured in your organization.

- **User-name mapping options**

- **Change the certificate mapping**

- **Address user-name ambiguity in multi-domain organizations**

User-name mapping options

Tableau Server uses one of the following approaches to map a client certificate to a user identity:

- **Active Directory.** If Tableau Server is configured to use Active Directory for user authentication, when Tableau Server receives a client certificate, it passes the certificate to Active Directory, which maps the certificate to an Active Directory identity. Any explicit user name information in the certificate is ignored.

  **Note:** This approach requires client certificates to be published for the user accounts in Active Directory.

- **User principal name (UPN).** A client certificate can be configured to store the user name in the user principal name field. Tableau Server reads the UPN value and maps it to a user in Active Directory or to a local user.

- **Common name (CN).** A client certificate can be configured to store the user name in the common name field of the certificate. Tableau Server reads the CN value and maps it to a user in Active Directory or to a local user.

If you configure the server for Active Directory authentication and UPN or CN user-name mapping, put the user name in one of the following formats:
username, domain/username, or username@domain.

For example: jsmith, example.org/jsmith, or jsmith@example.org.

If the server uses local authentication, the format of the name in the UPN or CN fields is not predetermined, but the name in the field must match a user name on the server.

Change the certificate mapping

You use the tsm authentication mutual-ssl <commands> commands to map a client certificate to a user identity in Tableau Server:

```
 tsm authentication mutual-ssl configure -m <value>
```

Possible values are ldap for Active Directory mapping, upn for UPN mapping, or cn for CN mapping.

When you first install and configure Tableau Server, the server sets the default user-name mapping to match the server’s authentication type:

- If the server is configured to use Active Directory, it also uses Active Directory for mapping the certificate to the user identity.
- If the server is configured to use local authentication, the server gets the user-name value from the UPN field in the certificate.

If the default behavior for how Tableau Server maps a user name to an identity is not correct for your server configuration, run the following set of commands to change the mapping to use the CN value:

```
 tsm authentication mutual-ssl configure -m cn

 tsm pending-changes apply
```

The pending-changes apply command displays a prompt to let you know this will restart Tableau Server if the server is running. The prompt displays even if the server is stopped, but in that case there is no restart. You can suppress the prompt using the --
ignore-prompt option, but this does not change the restart behavior. For more information, see tsm pending-changes apply.

Address user-name mapping ambiguity in multi-domain organizations

Under some circumstances, the user name in a certificate’s UPN or CN field can be ambiguous. This ambiguity can lead to unexpected results when the user name is mapped to a user identity on the server.

For example, if Tableau Server is presented with a user name that does not include a domain, the server maps the user name to an identity using the default domain. This can cause an incorrect user-name mapping, potentially assigning a user a different user’s identity and permissions.

This can occur particularly in environments where the following conditions apply:

- Your organization supports multiple Active Directory domains.
- The server is configured to use Active Directory authentication.
- The server is configured to use UPN or CN mapping.
- Some users have the same user name but different domains. For example, jsmith@example.org and jsmith@example.com.
- The user name in the certificate’s UPN or CN fields does not include the domain as part of the user name—for example, it shows jsmith.

To avoid incorrect user-name mapping, make sure the client certificates include fully qualified user names with the domain, using the format jsmith@example.org or example.org/jsmith.

How Mutual SSL Authentication Works

Mutual (or two-way) SSL authentication provides a combination of an encrypted data stream, mutual authentication of both server and client, and direct access convenience. To use mutual SSL with Tableau Server, you need the following:
- A trusted CA-issued SSL certificate for Tableau Server.
- A certificate on each client that will connect to Tableau Server.
- A Tableau Server configured to use mutual SSL.

Tableau Server and the client verify that each other has a valid certificate, and Tableau Server authenticates the user, based on the user name in the client certificate.

The following image shows a little more detail about the sequence of events that occurs with mutual SSL.

1. The user navigates to Tableau Server.
2. Tableau Server sends its SSL certificate to the client computer.
3. The client computer verifies the Tableau Server certificate.
4. The client computer sends its certificate to Tableau Server.
5. Tableau Server verifies the client certificate.
6. Tableau Server references the user name in the client certificate to authenticate the user.
System User, sudo Privileges, and systemd

This topic describes system user, systemd user service, and sudo privilege in the context of Tableau Server.

Privilege separation

Following standard security best practices, Tableau Server for Linux runs processes with the least privilege possible. During installation, the unprivileged user, *tableau*, is created in a server authorized group (*tableau*).

An example user entry in the /etc/passwd file is as follows:

- `tableau:x:993:991:Tableau Server:/var/opt/tableau/tableau_server:/bin/bash`

All processes run as the unprivileged *tableau* user. This means that if one of the Tableau Server processes (such as a process displaying vizzes to users) were compromised in some fashion, it would only be able to impact Tableau Server, not the rest of the Linux system. For this reason, you should not add the *tableau* unprivileged user to the *tsmadmin* group. The *tsmadmin* group should only contain accounts that require authorization to access OS-related Tableau configurations.

The *tableau* user and *tsmadmin* group are created by the Tableau Server initialization process. You can specify a different unprivileged user or TSM authorization group during installation. For more information about system users and groups, in the context of installation and LDAP configuration, see Identity Store.

*sudo* privileges

The first version (10.5) of Tableau Server on Linux relied on sudo privileges by updating the sudoers file. Updating the sudoers file conflicts with some system management configuration best practices and security policies. Therefore, the 2018.1 version (and later) of Tableau Server no longer creates or uses a privileged user (*tsmagent*). Nor does the current version of Tableau Server update or include a Tableau-specific sudoers file.
All privileged operations now occur during package and software installation.

**systemd user service**

In the 10.5 version of Tableau Server on Linux, sudo privileges were required to modify or restart the TSM services, which required systemctl commands. All TSM services were run from the normal system-wide systemd process (process ID 1, which runs all processes on the operating system). In this scheme, systemd process runs as root. Therefore, the 10.5 version of Tableau Server required sudo privileges.

With the current 2018.1 (and later) releases, we have removed the need for sudo privileges by making use of the systemd capability to run as a user service. The systemd user service runs as a normal user, so it does not need any special privileges once it has been enabled.

In normal use cases, you will not need to issue commands to systemd because TSM takes care of that. However, for troubleshooting scenarios, you may need to interact with the TSM services. As with the previous versions, you will issue the same systemctl commands for these scenarios. However, commands should be run as the `tableau` user, and not as root. If you specified a different unprivileged system user during Tableau Server setup, then run the commands as that user.

**Running systemctl commands**

Use the following syntax example to issue request to systemd with the systemctl commands.

Start a session as the unprivileged user. The `-l` flag is critical to set environment variables properly.

```
sudo su -l tableau
```

Then issue commands. For example:

```
systemctl --user status tabadmincontroller_0
systemctl --user restart tabadmincontroller_0
```
Security Hardening Checklist

The following list provides recommendations for improving the security ("hardening") of your Tableau Server installation.

1. Update to the current version

We recommend that you always run the latest version of Tableau Server. Additionally, Tableau periodically publishes maintenance releases of Tableau Server that include fixes for known security vulnerabilities. (Information regarding known security vulnerabilities can be found on the Security Bulletins page.) We recommend that you review maintenance release notifications to determine whether you should install them.

To get the latest version or maintenance release of Tableau Server, visit the Customer Portal page.

2. Configure SSL/TLS with a valid, trusted certificate

Secure Sockets Layer (SSL/TLS) is essential for helping to protect the security of communications with Tableau Server. Configure Tableau Server with a valid, trusted certificate (not a self-signed certificate) so that Tableau Desktop, mobile devices, and web clients can connect to the server over a secured connection. For more information, see SSL.

3. Disable older versions of TLS

Tableau Server uses TLS to authenticate and encrypt many connections between components and with external clients. External clients, such as browsers, Tableau Desktop, Tableau Mobile connect to Tableau using TLS over HTTPS. Transport layer security (TLS) is an improved version of SSL. In fact, older versions of SSL (SSL v2 and SSL v3) are no longer considered to be adequately secure communication standards. As a result, Tableau Server does not allow external clients to use SSL v2 or SSL v3 protocols to connect. We recommend that you only allow external clients to connect to Tableau Server with TLS v1.2.

Specifically, we recommend that you disable TLS v1 and TLS v1.1 on Tableau Server. However, before you disable a specific version of TLS, verify that the browsers that your
users connect to Tableau Server with support TLS v1.2. In some cases, you may need to preserve support for TLSv1.1.

The following tsm command enables TLS v1.2 (using the "all" parameter) and disables SSL v2, SSL v3, TLS v1, and TLS v1.1 (by prepending the minus [-] character to a given protocol).

```plaintext
tsm configuration set -k ssl.protocols -v 'all -SSLv2 -SSLv3 -TLSv1 -TLSv1.1'
tsm pending-changes apply
```

The `pending-changes apply` command displays a prompt to let you know this will restart Tableau Server if the server is running. The prompt displays even if the server is stopped, but in that case there is no restart. You can suppress the prompt using the `--ignore-prompt` option, but this does not change the restart behavior. For more information, see `tsm pending-changes apply`.

4. Configure SSL encryption for internal traffic

Configure Tableau Server to use SSL to encrypt all traffic between the Postgres repository and other server components. By default, SSL is disabled for communications between server components and the repository. We recommend enabling internal SSL for all instances of Tableau Server, even single-server installations. Enabling internal SSL is especially important for multi-node deployments. See Configure SSL for Internal Postgres Communication.

5. Enable firewall protection

Tableau Server was designed to operate inside a protected internal network.

**Important:** Do not run Tableau Server, or any components of Tableau Server on the internet or in a DMZ. Tableau Server must be run within the corporate network.
protected by an internet firewall. We recommend configuring a reverse proxy solution for internet clients that need to connect to Tableau Server. See Configuring Proxies for Tableau Server.

A local firewall should be enabled on the operating system to protect Tableau Server in single and multi-node deployments. In a distributed (multi-node) installation of Tableau Server, communication between nodes does not use secure communication. Therefore, you should enable firewalls on the computers that host Tableau Server. See Configure Local Firewall.

To prevent a passive attacker from observing communications between nodes, configure a segregated virtual LAN or other network layer security solution.

See Tableau Services Manager Ports to understand which ports and services Tableau Server requires.

6. Restrict access to the server computer and to important directories

Tableau Server configuration files and log files can contain information that is valuable to an attacker. Therefore, restrict physical access to the machine that is running Tableau Server. In addition, make sure that only authorized and trusted users have access to the Tableau Server files in the /var/opt/tableau/tableau_server/ directory.

7. Generate fresh secrets and tokens

Any Tableau Server service that communicates with repository or the cache server must first authenticate with a secret token. The secret token is generated during Tableau Server setup. The encryption key that internal SSL uses to encrypt traffic to Postgres repository is also generated at during setup.

We recommend that after you install Tableau Server, you generate new encryption keys for your deployment.
These security assets can be regenerated with the `tsm security regenerate-internal-tokens` command.

Run the following commands:

```bash
tsm security regenerate-internal-tokens

tsm pending-changes apply
```

8. Disable services that you're not using

To minimize the attack surface of the Tableau Server, disable any connection points that are not needed.

REST API

The REST API interface is enabled by default. If no applications will make REST API calls to your installation of Tableau Server 9.3 (or later), disable it by using the following commands:

```bash

tsm configuration set -k api.server.enabled -v false

tsm pending-changes apply
```

**Important:** Tableau Prep uses REST API to access Tableau Server. If your organization uses Tableau Prep, do not disable REST API.

JMX Service

JMX is disabled by default. If it's enabled but you're not using it, you should disable it by using the following:

```bash

tsm configuration set -k service.jmx_enabled -v false

tsm pending-changes apply
```
9. Verify session lifetime configuration

By default, Tableau Server does not have an absolute session timeout. This means that client sessions can remain open indefinitely if the Tableau Server inactivity timeout is not exceeded. (The default inactivity timeout is 240 minutes.)

If your security policy requires it, you can set an absolute session timeout. Be sure to set your absolute session timeout in a range that allows the longest-running extract or publishing operations in your organization. Setting the session timeout too low may result in extract and publishing failures for long-running operations.

To set the session timeout run the following commands:

```
tsm configuration set -k wgserver.session.apply_lifetime_limit -v true
```
```
tsm configuration set -k wgserver.session.lifetime_limit -v value, where value is the number of minutes. The default is 1440, which is 24 hours.
```
```
tsm configuration set -k wgserver.session.idle_limit -v value, where value is the number of minutes. The default is 240.
```
```
tsm pending-changes apply
```

10. Configure a server safelist for file-based data sources

By default, Tableau Server allows authorized Tableau Server users to build workbooks that use files on the server as file-based data sources (such as spreadsheets). In this scenario, files are accessed by the `tableau` system account.

To prevent unwanted access to files, we recommend that you configure safelist (sometimes referred to as "whitelist") functionality. This lets you limit `tableau` account access to just the directory paths where you host data files.
1. On the computer running Tableau Server, identify the directories where you will host data source files.

   **Important** Make sure the file paths you specify in this procedure exist on the server. If the paths do not exist when the computer starts, Tableau Server will not start.

2. Run the following commands:

   ```bash
   tsm configuration set -k native_api.allowed_paths -v "path", where path is the directory to add to the safelist. All subdirectories of the specified path will be added to the safelist. If you want to specify multiple paths, separate them with a semicolon, as in this example:

   tsm configuration set -k native_api.allowed_paths -v "/datasources;/HR/data"

   tsm pending-changes apply
   ```

11. Enable HTTP Strict Transport Security for web browser clients

   HTTP Strict Transport Security (HSTS) is a policy configured on web application services, such as Tableau Server. When a conforming browser encounters a web application running HSTS, then all communications with the service must be over a secured (HTTPS) connection. HSTS is supported by major browsers.


   To enable HSTS, run the following commands on Tableau Server:

   ```bash
   tsm configuration set -k gateway.http.hsts -v true
   ```
By default, HSTS policy is set for one year (31536000 seconds). This time period specifies the amount of time in which the browser will access the server over HTTPS. You should consider setting a short max-age during initial roll-out of HSTS. To change this time period, run
tsm configuration set -k gateway.http.hsts_options -v max-age-e=<seconds>. For example, to set HSTS policy time period to 30 days, enter tsm configuration set -k gateway.http.hsts_options -v max-age=2592000.

tsm pending-changes apply

12. Disable Guest access
Core-based licenses of Tableau Server include a Guest user option, which allows any user in your organization to see and interact with Tableau views embedded in web pages.

Guest user access is enabled by default on Tableau Servers deployed with core-based licensing.

Guest access allows users to see embedded views. The Guest user cannot browse the Tableau Server interface or see server interface elements in the view, such as user name, account settings, comments, and so on.

If your organization has deployed Tableau Server with core licensing and Guest access is not required, then disable Guest access.

You can disable Guest access at the server or site level.

You must be a server administrator to disable the Guest account at either the server or the site level.

To disable Guest access at the server level:

1. In the site menu, click Manage All Sites and then click Settings > General.

2. For Guest Access, clear the Enable Guest account check box.

3. Click Save.
To disable Guest access for a site:

1. In the site menu, select a site.

2. Click Settings, and on the Settings page, clear the Enable Guest account check box.

For more information, see Guest User.

13. Set referrer-policy HTTP header to 'same-origin'

Beginning in 2019.2, Tableau Server includes the ability to configure Referrer-Policy HTTP header behavior. This policy is enabled with a default behavior that will include the origin URL for all "secure as" connections (no-referrer-when-downgrade), which sends origin referrer information only to like connections (HTTP to HTTP) or those that are more secure (HTTP to HTTPS).

However, we recommend setting this value to same-origin, which only sends referrer information to same-site origins. Requests from outside the site will not receive referrer information.

To update the referrer-policy to same-origin, run the following commands:

```
tsm configuration set -k gateway.http.referrer_policy -v same-origin
```

```
tsm pending-changes apply
```

For more information about configuring additional headers to improve security, see HTTP Response Headers.

Change List

<table>
<thead>
<tr>
<th>Date</th>
<th>Change</th>
</tr>
</thead>
<tbody>
<tr>
<td>September 2017</td>
<td>Ported and updated for Tableau Services Manager and Linux platform.</td>
</tr>
</tbody>
</table>
Manage Licenses

You can manage your Tableau Server licenses and view license usage.

Licensing Overview

Tableau Server term licenses are available with two different license metrics: User-Based and Core-Based. Term licenses, also known as subscription licenses, allow you to use and update Tableau Server for a specified period of time.

Tableau offers multiple types of User-Based term licenses that grant a range of capabilities at various price points, providing the flexibility for organizations to pay for the data analysis and data visualization capabilities that each type of user in their organization needs. To learn more, see User-based licenses.

License metric: User-Based or Core-Based

Your license is defined by the metric that permits use of Tableau Server.

- A User-Based license metric allows you to deploy Tableau Server on a single computer or on multiple computers in a cluster. Each user that accesses Tableau Server must be licensed. Administrators add users and license them.

- A Core-Based license metric imposes no constraints on the number of user accounts
in Tableau Server. Instead, the license specifies the maximum number of computer cores on which you can run Tableau Server. You can install Tableau Server on a single computer or across multiple computers as a multi-node cluster, as long as the total number of cores in all the computers does not exceed the total number that the license allows.

**Note:** Not all processes installed with Tableau Server impact the calculation of total number of cores used. A subset of processes are considered "licensed processes." Core licensing is calculated only on computers running licensed processes. If a computer has one or more licensed processes installed on it, the cores on that computer count toward the total cores used. For more information about licensed processes, see Licensed processes.

The topics in this section provide guidance about how to view and refresh Tableau licensing, as well as a topic that describes how to add user capacity on Tableau Server.

**Note:** As a Tableau Server administrator, you may also be tasked with managing Tableau Desktop deployment and tracking license usage on client computers. If your organization will be using Desktop License reporting, you need to configure Tableau Server to support this. For details, see Configure Desktop License Reporting. For additional information on licensing and management tasks related to Tableau Desktop, see the Tableau Desktop Deployment Guide.

**User-Based licenses**

Tableau offers User-Based term licenses that grant a range of capabilities at various price points. Multiple types of User-Based term licenses are available: Viewer, Explorer and Creator. The following capabilities are available to users with these licenses:

- Viewer licenses let users view and interact with workbooks in Tableau Server or Tableau Online. Viewer licenses also let users access Tableau Mobile, add
comments to workbooks, download .png and .pdf versions of workbooks, download workbook summary data, create subscriptions for themselves, and receive data-driven alerts.

- Explorer licenses are equivalent to the User-Based term licenses available in previous Tableau Server and Tableau Online releases, and include the capabilities provided with Viewer licenses. Depending on a user’s site role, a user with this license will have access to workbook authoring capabilities using a web browser, as well as a full set of collaboration features.

- Creator licenses grant a wide range of capabilities when using Tableau Server and Tableau Online, and also grant use of Tableau Desktop and Tableau Prep. Depending on a user’s site role, a user with a Creator license might have all of the capabilities available under the Explorer license, as well as the following capabilities when using Tableau Server and Tableau Online:
  - Create and publish new workbooks from a new data source.
  - Edit embedded data sources in the Data pane.
  - Create and publish new data connections.
  - Create new workbooks using Dashboard Starters (Tableau Online only).

**Note:** Tableau Server administrators will always use a Creator license, if available; additionally, activating a Creator license on Tableau Server causes all Tableau Server administrator accounts to require Creator licenses.

Product keys provide administrators with a set of each type of license, which they can activate by assigning associated site roles to users. The first Creator or Explorer license that you add to Tableau Server activates Tableau Server, and will be used by a Server Administrator. Administrators can activate additional Creator, Explorer and Viewer licenses in Tableau Server or Tableau Online to increase licensed user capacity.

**Data Management Add-On**

The Data Management Add-on product key includes Tableau Prep Conductor for a single Tableau Server Deployment, which may be User-Based or Core-Based. A Deployment includes a licensed production Tableau Server installation and licensed non-production
Tableau Server installations that support the production installation. For more information, see Use the Data Management Add-on.

Tableau Server Management Add-on

The Tableau Server Management Add-on is licensed on a per Deployment basis, which may be User-Based or Core-Based. A Deployment includes a licensed production Tableau Server installation and two licensed non-production Tableau Server installations that support the production installation. For more information on Server Management Add-on and the features included in this Add-on, see About Tableau Server Management Add-on.

Tableau Server licenses and site roles after an upgrade

When upgraded to 2018.2, Tableau Server will display the new licensing information and site roles. Any users in the Read Only site role prior to the upgrade are reassigned to the Viewer site role.

To learn more about the site roles available with each type of User-Based license, see Set Users’ Site Roles. To learn about pricing for each type of User-Based license, see Tableau Pricing. To learn about troubleshooting issues with licensing, see Troubleshoot Licensing.

Tableau Server product key activation

You can activate the same Tableau Server product key up to three times. This allows you to test Tableau Server (in a sandbox or QA environments, for example), as well as use Tableau in production. To maximize your activations, you should deactivate your product key when you remove Tableau Server from a computer, unless you will be reinstalling Tableau on the same computer. Doing this gives you the opportunity to use the activation on a different computer. For example, if you move Tableau Server from one computer to another, deactivate the product key, then remove Tableau from the original computer.

When you install Tableau on the new computer, you can activate the key there without any conflict. If you are removing Tableau Server to reinstall it on the same computer, you don't need to deactivate the key. Tableau will use the key when reinstalled. For example, if you are moving Tableau from one drive on a computer to a different drive on the same computer. For information on how to deactivate a product key, see tsm licenses deactivate.
Tableau Server licensing and virtual machines (VMs)

If you run Tableau Server on VMs, either locally, or in the cloud, be aware of the potential for complications related to licensing. If you are simply upgrading Tableau Server on the VM, you do not need to take any extra action related to licensing. If you plan to clone the VM to create either a new production or test environment to upgrade, you need to deactivate any Tableau Server licenses before cloning. If you do not do this, the new VM environment can end up with an untrusted license, and any attempts to upgrade will fail. You may also end up hitting the maximum number of activations for the licenses.

To avoid issues with licensing on VMs, deactivate all Tableau licenses before cloning a VM or allowing it to be permanently shut down.

View Server Licenses

Server administrators can view the license and product key information for Tableau Server.

Viewing licenses from the Tableau Server web UI

How you navigate to the Licenses page in Tableau Server depends on whether you have a single site, or multiple sites.

- On a server with a single site, click **Settings and Licenses**:

- On a multi-site server, click **Manage all sites** on the site menu, **Settings**, and **Licenses**:

  **Note:** The **Manage all sites** option only displays when you are signed in as a server administrator.

This page displays information for any licenses that have been activated on your server, including any user-based (term) or core-based licenses.

Use the TSM web interface
1. Open TSM in a browser:

   http://<tsm-computer-name>:8850

2. Click **Configuration**, and then click **Licensing**:

   The table displays the product key, expiration date, and expiration of maintenance.

   **Note:** The TSM Web UI provides a limited amount of licensing information. Use the TSM CLI or the Tableau Server Web UI to see additional licensing information, including the number of each type of user-based license (Creator, Explorer and Viewer).

**Use the TSM CLI**

1. Open a command prompt as administrator on the initial node (the node where TSM is installed).

2. Run the following command:

   `tsm licenses list`

   The output reflects whether Tableau Server uses a user-based license, a core-based license, or a combination of the two. The output of this command is formatted as follows:

<table>
<thead>
<tr>
<th>KEY</th>
<th>TYPE</th>
<th>CREATOR</th>
<th>EXPLORER</th>
<th>VIEWER</th>
<th>GUEST ACCESS</th>
<th>LIC EXP</th>
<th>MAINT EXP</th>
<th>VALID</th>
</tr>
</thead>
<tbody>
<tr>
<td>KE- Y</td>
<td>Term</td>
<td>Seat #</td>
<td>Seat #</td>
<td>Seat #</td>
<td>true</td>
<td>mm/dd/yy</td>
<td>mm/dd/yy</td>
<td>true</td>
</tr>
<tr>
<td>KE- Y</td>
<td>Core- s</td>
<td></td>
<td></td>
<td></td>
<td>false</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
For example, a server with a core-based license and a second product key that grants user-based licenses might provide command output similar to the following:

<table>
<thead>
<tr>
<th>KEY</th>
<th>TYPE</th>
<th>CREATOR</th>
<th>EXPLORER</th>
<th>VIEWER</th>
<th>GUEST ACCESS</th>
<th>LIC EXP</th>
<th>MAINT EXP</th>
<th>VALID</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;KE-Y&gt;</td>
<td>Term</td>
<td>5</td>
<td>5</td>
<td>100</td>
<td>false</td>
<td>mm/d-d/yy</td>
<td>N/A</td>
<td>true</td>
</tr>
<tr>
<td>&lt;KE-Y&gt;</td>
<td>32 Cor-</td>
<td>0</td>
<td>Unlim-</td>
<td>Unlim-</td>
<td>true</td>
<td>N/A</td>
<td>mm/d-d/yy</td>
<td>true</td>
</tr>
<tr>
<td>&lt;KE-Y&gt;</td>
<td>32 Cor-</td>
<td>0</td>
<td>Unlim-</td>
<td>Unlim-</td>
<td>true</td>
<td>N/A</td>
<td>mm/d-d/yy</td>
<td>true</td>
</tr>
</tbody>
</table>

Note: The license terms for Creator, Explorer and Viewer users are set according to the terms of the user-based license (term license), if present. So, a server with only a core-based license will have unlimited Explorer and Viewer users and guest access, but no Creator users. Adding a user-based license overrides these settings by providing a specific number of Creator, Explorer and Viewer users, and disabling guest access. To learn more, see Use user-based licenses on a server with core-based licensing.

Refresh Maintenance Date for the Product Key

When you purchase a new subscription, you must refresh your product key(s). Refreshing the product key(s) updates the product key with the new date that reflects the most up-to-date maintenance and expiration dates. If the product key is not refreshed and has expired, Tableau will stop working and you will have to activate a new product key. If you believe that the latest displayed date is incorrect, contact Tableau Support.
Note: This topic describes how to refresh the maintenance date for Tableau Server. For information about refreshing the maintenance date on Tableau Desktop see Refreshing Tableau Desktop Product Key in the Tableau Knowledge Base.

Use the TSM web interface

1. Open TSM in a browser:
   
   http://<tsm-computer-name>:8850

2. Click Configuration and Licensing and click Refresh All:

Use the TSM CLI

1. Open a command prompt as administrator on the initial node (the node where TSM is installed).

2. Run the following command:
   
   tsm licenses refresh
Add Capacity to Tableau Server

You may need to add capacity to your Tableau Server installation to allow you to increase the number of users (if you have a user-based license) or the number of cores (if you have a core-based license).

Tableau Software will provide you with a new product key that adds capacity to your existing Tableau Server installation. You need to activate this key and use it together with your existing product key(s) to get the combined capacity you are licensed for.

Follow the steps below to add a product key to Tableau Server.

**Note:** This process requires a restart of Tableau Server.

Use the TSM web interface

1. Open TSM in a browser:

   https://<tsm-computer-name>:8850

2. Click **Licensing** on the **Configuration** tab and click **Activate License**:

3. Enter or paste your new product key and click **Activate**:
4. On the Register page, enter your information into the fields and click **Register**.
Register with Tableau. All fields are required.

**Contact Information**
- First Name
- Last Name
- Phone Number
- Email

**Company Information**
- Organization
- Industry
- Department
- Job Role

**Region Information**
- City
- Postal Code
- Country/Region
- State/Province

Register
5. Restart Tableau Server after registration is complete.

Use the TSM CLI

1. Copy the product key to your computer.

2. Run the following command:

   ```bash
   tsm licenses activate --license-key <license key>
   ```

3. After activation, you need to register Tableau Server. To do this, create a registration file and then pass it as an option with the `tsm register` command.
   
   a. Generate a template that you can edit by running the following command:

   ```bash
   tsm register --template > /path/to/<registration_file>.json
   ```

   b. Edit the template file to create your completed registration file.

   Here is an example of a registration file in the required format:

   ```json
   {
     "zip" : "97403",
     "country" : "USA",
     "city" : "Springfield",
     "last_name" : "Simpson",
     "industry" : "Energy",
     "eula" : "yes",
     "title" : "Safety Inspection Engineer",
     "phone" : "5558675309",
     "company" : "Example",
     "state" : "OR",
     "department" : "Engineering",
     "first_name" : "Homer",
   }
   ```
c. After saving changes to the file, pass it with the --file option to register Tableau Server:

```bash
tsm register --file /path/to/<registration_file>.json
```

For example:

```bash
tsm register --file /usr/share/tableau-reg-file.json
```

4. Restart the server:

```bash
tsm restart
```

### Activate Tableau Server Offline - Adding a License

When you install Tableau Server, you have to activate at least one product key. Doing this activates the server, and specifies the number of license levels you can assign to users. There are also times you may need to activate licenses after Tableau Server is installed, for example, if you add capacity to your server, or get a new product key. If you don’t have your product key, you can get it from the Tableau Customer Account Center.

In most cases, you can activate your key directly from Tableau, either during installation, or later, using the Tableau Services Manager (TSM) Licenses page, but there are some circumstances that don’t allow you to do this. If your computer is not connected to the internet for example, or has a firewall that restricts access outside your intranet. In these cases you need to do an offline activation.

- Offline activation during install—To complete an offline activation when you are installing Tableau Server, see Offline Activation When Installing Tableau Server.
- Offline activation of licenses after install—To complete an offline activation after your server is installed and running, follow the steps below.
Offline activation overview and prerequisites

Offline activation of Tableau Server involves several steps. If you have never activated a product key before on your computer, you need to run through the steps twice, once to initialize licensing on the computer, and a second time to activate the key. If you have had a license on the computer in the past, you run through the steps only once.

Offline activation steps:

1. Generate an offline activation request file (`offline.tld`).
2. Upload the offline activation request to Tableau and download the resulting activation file (`activation.tlf`).
3. Upload the activation file to Tableau Server.

If this is the first time a Tableau product key has been activated on the computer, you repeat these three steps a second time.

Use the TSM web interface

If you attempt to activate your product key from the TSM licenses page and see a dialog that says online activation is unavailable, you can activate the key offline. Click Activate License Offline.
1. Create Offline File

Your first step is to create an offline file (offline.tlq).

Create an offline file you will upload to Tableau. If your license key is not pre-filled in the form, enter your key and click Create Offline File to generate an offline file (offline.tlq) on the local computer.

Copy the offline.tlq file to a computer with internet access. You need to upload this file to Tableau to generate an activation file.

2. Upload and Submit Offline File

Next, on the computer where you copied the offline.tlq file, go to http://www.tableau.com/support/activation to upload your file and submit it. Doing this automatically generates an activation file (activation.tlf) that you can download and copy back to the Tableau Server computer.

a. On the Offline Activation page, click Choose File to select the offline.tlq file.
b. Click **Upload Activation File** to submit the file to Tableau.

c. Click the [here](#) link to download the resulting `activation.tlf` file to your computer.

**Offline Activation**

The activation was successful. Please click [here](#) to download your activation file.

For help creating the offline activation file, see [Activate Tableau Desktop Offline](#) or [Activate Tableau Server Offline (Linux)](#).

d. Copy the downloaded `activation.tlf` file to the computer where Tableau Server is installed.

3. Click **Next Step** to move to the next dialog.

4. Upload Activation File

On the Tableau Server computer, click **Upload Activation File** to upload the `activation.tlf` file to Tableau Server.
5. Create, Upload, and Submit a Second Offline File
   
   a. Click Create Offline File to generate a second offline.tlq file.
   b. Copy that file to the computer that has internet access.
   c. Open a browser session to http://www.tableau.com/support/activation and upload your new offline file.
   d. Download the resulting activation.tlf file.
   e. Copy the file to the Tableau Server computer.

6. Click Next Step and then click Upload Activation File to upload the second activation.tlf file to Tableau Server.
7. Click **Activate License** to complete the offline activation. You will need to restart Tableau Server for the changes to take effect.

**Use the TSM CLI**

Log in to Tableau Services Manager

Before you can proceed you must log in to Tableau Services Manager (TSM).

To log in to TSM, run the following command:

```
 tsm login -u <username>
```

**What if I can't log in?**

The account that you use to configure the rest of the installation must be a member of the `tsadmin` group that was created during initialization. To view the user accounts in the `tsadmin` group, run the following command:

```
 grep tadmin /etc/group
```

If the user account is not in the group, run the following command to add the user to the `tsadmin` group:

```
 sudo usermod -G tadmin -a <username>
```

After you have added the user to the `tsadmin` group, run the `tsm login` command.

**Step 1 - Generate an offline activation request file**

1. On the initial node, open a terminal session.

2. Type this command to get your offline activation file:

   ```
   tsm licenses get-offline-activation-file -k <product-key> -o <target-directory>
   ```
You can get your product key from the Tableau Customer Portal. The target directory must already exist.

3. Copy the offline activation file (offline.tlq) from the target directory to a computer that has internet access.

Step 2 - Upload the offline activation request to Tableau

1. On the computer that has internet access, go to the Tableau Product Activations page.

2. Complete the instructions to upload your offline.tlq file.

   This creates an activation file, activation.tlf.

3. Download the resulting activation file from Tableau.

Step 3 - Initialize or activate your license

1. Copy the activation file (activation.tlf) to a location accessible from your Tableau Server computer.

2. Run the following command:

   tsm licenses activate -f <path-and-activation-file>

3. Restart the server for licensing changes to take effect:

   tsm restart

First time license activation

If this is the first Tableau license activated on the computer, you will see this message:

Your license has been initialized. To complete the activation, we need one more exchange. Generate and send to Tableau a second activation request file.
Repeat steps 1 through 3 above to activate your license. You need to generate a second offline.tlg file, upload it to Tableau, and download the resulting activation.tlf file, then use that file to activate your license.

If you are not sure whether you need to repeat the steps, you can run this command:

tsm licenses list

If you see a message like this one you need to repeat the steps above:

No licenses are currently activated.

Subsequent license activation

If this is not the first Tableau license activated on the computer, or if you have completed the steps above two times, you should see a success message:

Activation successful.

Tableau Server is activated. If you need additional assistance, contact Tableau Technical Support.

Automate Licensing Tasks

Use the tsm licenses command to automate licensing tasks.

C:\ProgramData\Tableau\Tableau Server\data\tabsvc\logs\

Troubleshoot Licensing

This topic includes instructions for troubleshooting issues related to Tableau Server licensing.

Handle an unlicensed server

Tableau offers two licensing models: user-based and core-based. To learn more about user-based and core-based licensing, see Licensing Overview.
User-based licensing requires each active user account to be covered by a license. User-based licenses have a defined capacity, or number of users that they allow. Each user is assigned a unique user name on the server and is required to identify themselves when connecting to the server.

Core-based licensing has no constraints on the number of user accounts in the system, but it does restrict the maximum number of processor cores that Tableau Server can use. You can install Tableau Server on one or more machines to create a cluster, with the restriction that the total number of cores in all the machines does not exceed the number of cores you have licensed and that all of the cores on a particular machine are covered by the license.

Unlicensed User-Based Server

The most common reason for a server that has user-based licensing to be unlicensed is an expired product key or an expired maintenance contract.

Unlicensed Core-Based Server

A core-based server can become unlicensed for a variety of reasons, such as when Tableau Server nodes running licensed processes cannot contact the Tableau Server node running the License Manager service. To learn more about licensed processes, see Tableau Server Processes.

When the server is unlicensed you may not be able to start or administer the server. You can, however, manage your licenses using the tsm licenses command.

Troubleshoot user-based licensing

This section provides information about resolving issues that can occur when adding the user-based Viewer, Explorer and Creator licenses to Tableau Server or Tableau Online, or when these licenses expire. The highest available license type is Creator, followed by Explorer, and finally Viewer. To learn more about user-based licensing, see Licensing Overview.
A user or administrator is unlicensed due to license expiration.

To avoid having users unexpectedly become unlicensed or move to another site role, you should always do one of the following before the license that they are currently using expires:

- Renew and activate a replacement license. If a user occupies a Creator, Explorer or viewer license and their license expires, they will use another license of the same type, if available.
- Change the site role of those users to allow the use of a license that is not due to expire.

To learn how site roles can be changed to require a different license, see Set Users' Site Roles.

The reassignment of users to new licenses is governed by the following logic:

- When a Server Administrator user occupies a Creator license and their license expires (with no replacement licenses available), they are reassigned to an Explorer license if any Explorer licenses are available. This license reassignment occurs in order of most recent login. Server Administrators displace other users who might be currently using an Explorer license. If no Creator or Explorer licenses are available a Server Administrator becomes unlicensed.
- When a non-Server Administrator user occupies a Creator license and their license expires (with no replacement licenses available), they become unlicensed. To avoid having these users become unlicensed, change their site role prior to license expiration. This is especially important for users in the Site Administrator Creator site role, who must move to the Site Administrator Explorer site role before their Creator license expires to avoid losing Site Administrator capabilities.
- When a non-Server Administrator user occupies an Explorer or Viewer license and their license expires (with no replacement licenses available), they are upgraded to a higher license type, if licenses of that type are available. Specifically, the following occurs when a license expires:
  - Users who occupy an Explorer license will move to a Creator license, if available (with no change to site role).
Users who occupy a Viewer license will move to an Explorer license, if available. If no Explorer licenses are available, these users will move to a Creator license, if available (with no change to site role). If no licenses are available at the higher license types, those users are moved to Unlicensed.

Users are reassigned to a new license as described above in order of most recent login, with lower license types reassigned first (first Viewer, then Explorer, and then Creator).

For example: Two users with a Viewer license, a user with the Creator license, and two Server Administrators with a Creator license all have their licenses expire. Four unexpired Explorer licenses are available for these users. In this situation, the following occurs in the order shown below:

1. The user with a Viewer license who logged in most recently is reassigned to an Explorer license.
2. The second user with a Viewer license is reassigned to an Explorer license.
3. The Server Administrator user with a Creator license who logged in most recently is reassigned to an Explorer license, and then the second Server Administrator with a Creator license is reassigned to the remaining Explorer license.
4. The user with the Creator license becomes unlicensed.

Server Administrator site role is unchanged when using a Creator license

Server Administrators gain Creator capabilities if Creator licenses are available in Tableau Server, with no change to their site role name. All other Tableau Server and Tableau Online users gain Creator licenses only if assigned to a site role that includes Creator in its name.

Licenses are not immediately available

When you add a role-based license to Tableau Server, those licenses become available to all users when you restart Tableau Server. If you don’t restart Tableau Server, role-based licenses become available to all users within 24 hours.
A user with a Viewer license cannot open Tableau Server or Tableau Online workbooks from Tableau Desktop.

A user with a Viewer license who also has a separate Tableau Desktop license will be unable to open workbooks on Tableau Server or Tableau Online using Tableau Desktop. To open workbooks such using Tableau Desktop, that user will need an Explorer or Creator license on Tableau Server or Tableau Online.

Migrate from Core-Based to User-Based Licensing

This guide tells you how to migrate Tableau Server from a core-based license metric (which counts the processor cores on which you have Tableau Server installed) to a user-based license metric (which counts named users). To learn more about licensing metrics, see Licensing Overview.

**Note:** As an alternative to migrating to user-based licensing, you can also continue using a core-based license and add user-based Viewer, Explorer and Creator licenses as-needed to provide users with the capabilities granted by these licenses. To learn more, see Use user-based licenses on a server with core-based licensing.

Prepare for migration to user-based licensing

Core-based licenses allow an unlimited number of users, including view-only guest accounts. Every user has a site role when they are added to Tableau Server, and these users and site roles persist when licensing is changed. Because user-based licenses limit the number of users, you should ensure that your new user-based licenses accommodate the number of users who are currently connecting to Tableau Server, including the users currently using guest accounts. If your new user-based licenses don't accommodate the full number of users, some users will move to the Unlicensed site role. To learn more about site roles, see Set Users' Site Roles.
To count the number of users in your Tableau Server installation, export a list of users to count them with a tool such as Microsoft Excel. To learn how to export a list of users, see Export a User List.

Migrate to user-based licensing

To migrate to user-based licensing you must stop Tableau Server, deactivate the core-based product key, activate the user-based product key(s), and then start Tableau Server. Because this process will result in a restart of Tableau Server and cause downtime for Tableau Server users, you should migrate licensing during a period of low usage.

1. Stop Tableau Server:

   Use the `tsm stop` command. To learn more, see: `tsm stop`.

2. Deactivate the core-based product key:

   Use the `tsm licenses deactivate` command with the core-based product key.

3. Activate the user-based product key:

   Use the `tsm licenses activate` command with the user-based product key.

4. Start Tableau Server:

   Use the `tsm start` command.

Use user-based licenses on a server with core-based licensing

The 2018.1 release of Tableau Server now lets you to add user-based licenses to Tableau Server installations with core-based licensing. If you upgrade a Tableau Server installation to 2018.1 without activating user-based licenses, Tableau Server will continue to operate as it did previously, with no changes to Tableau Server UI or permissions except that the legacy Viewer site role is renamed to Read Only.

The first Creator or Explorer license that you add to Tableau Server will be used by Server Administrator users. Administrators can activate additional Creator, Explorer and Viewer licenses in Tableau Server to increase licensed user capacity. Activating these user-based licenses gives you a combination of the capabilities granted by user-based Creator, Explorer
and Viewer licenses and the capabilities granted by your core-based license. To learn more about the different types of user-based licenses, see User-based licenses.

**Note:** The unlimited number of users who have full access to Tableau Server under core-based licensing have equivalent capabilities to users with an Explorer license under user-based licensing.

For example, if a Tableau Server installation has a 16 core license that includes guest access, and you added 10 Creator licenses, that server would have the following capabilities:

- A limit of 16 processor cores on hardware that runs Tableau Server
- Guest access
- Unlimited Explorer licenses (from the unlimited user licenses that come with a core license)
- 10 Creator licenses

Example of completing a migration from core-based licensing

To extend the example above: If the core-based license was then deactivated, the following capabilities would be available:

- No limits on server hardware
- No guest access
- 10 Creator licenses

If you then added 50 Explorer licenses and 200 Viewer licenses, the following capabilities would be available:

- No limits on server hardware
- No guest access
- 10 Creator licenses
- 50 Explorer licenses
- 200 Viewer licenses
Login-based License Management

**Note:** Login-based license management is currently offered in a limited release to selected customers. Login-based license management requires a special Tableau Server product key. Contact your Tableau sales representative if you are interested in using login-based license management.

Login-based license management, helps you manage licensing for users with Creator roles on Tableau Server and Tableau Online. If you’re using Role Based Subscriptions with Tableau Server or Tableau Online, you can simplify your license management using login-based license management to eliminate separate Tableau Desktop and Tableau Prep Builder product keys. You only need to manage one or more product keys for on-premises Tableau Server, or in the case of Tableau Online, you don’t need to manage any product keys at all.

You only need one Tableau Server or Tableau Online site to authorize an individual Tableau Desktop or Tableau Prep Builder. Login-based license management is enabled per production instance of your Server deployment. You can assign Creator roles to users who use Tableau Server, Tableau Desktop, and Tableau Prep Builder for license activation and centralized license management. In addition, login-based license management gives you more visibility into license usage. You can use administrative views on Tableau Server and Tableau Online to see the assignment and use of Creator seats in one place. Administrative views provide information about the most recent license lease and version of both Tableau Desktop and Tableau Prep Builder to help you monitor your Tableau deployments.

**Use login-based license management**

To use login-based license management, you either need to use Tableau Online or you need to install Tableau Server and activate it with a role-based subscription Creator product key. You will also need to install Tableau Desktop.
Step 1: Install Tableau Server

We recommend removing previous versions of Tableau Server from your computer. See Remove Tableau Server from Your Computer.

1. Refer to the documentation in the Deploy chapter to install Tableau Server.

2. During activation, you must enter a Server product key that has been created to support role based subscription and login-based license management.

3. Add users to your Tableau Server and set them to one of the three Creator roles: Server Administrator, Site Administrator Creator, or Creator. This assigns a Creator license to these users on Tableau Server. For more information, see Set User’s Site Roles.

Step 2: Enable login-based license management

Login-based license management is an opt-in feature (it is disabled by default), so you need to enable the feature in order to use it. You can use the following steps to enable login-based license management:

Enable login-based license management on Tableau Server

On Tableau Server, you use the Tableau Services Manager (TSM) command line utility to turn on or turn off login-based license management. In addition, you can use the tsm command line utility to set the ATR duration.

To turn on login-based license management, at a command prompt, type:

```
tsm configuration set -k licensing.login_based_license_management.enabled -v true
tsm pending-changes apply
```

To turn off login-based license management, at a command prompt, type:
Tableau Server on Linux Administrator Guide

tsm configuration set -k licensing.login_based_license_management.enabled -v false

tsm pending-changes apply

To set the ATR duration, at a command prompt, type:

tsm configuration set -k licensing.login_based_license_management.default_requested_duration_seconds -v <value in seconds>

tsm pending-changes apply

To set the ATR maximum duration, at a command prompt, type:

tsm configuration set -k licensing.klicensing.login_based_license_management.max_requested_duration_seconds -v <value in seconds>

tsm pending-changes apply

Enable login-based license management on Tableau Desktop at install

Run the installer .exe file from your computer's command line as an administrator, including the LBLM=enabled switch. If you need to extract the .msi files, follow the instructions to Extract and run the Windows (MSI) installer.

To use a duration length other than the default of 14 days, include the ATRREQUESTEDDURATIONSECONDS switch. For example:

tableauDesktop-64bit-2019-3-0.exe /quiet /norestart ACCEPTEULA=1 ATRREQUESTEDDURATIONSECONDS=43200
You must run the command from the directory where the .exe file is located or specify a full path to the location of the .exe file on the computer. Do not run the setup program from a shared directory on your network. Instead, download the .exe file to a directory on the computer where you're installing.

The following example shows the Windows installer command that enables login-based license management:

```
tableauDesktop-64bit-2019-3-0.exe /quiet /norestart
ACCEPTEULA=1 LBLM=enabled
```

To set the default URL for the Tableau Server you want users to use for activation when using login-based license management, add the `ACTIVATIONSERVER` option:

```
tableauDesktop-64bit-2019-3-0.exe /quiet /norestart
ACCEPTEULA=1 ATRREQUESTEDDURATIONSECONDS=43200 ACTIVATIONSERVER-R=http://<tableau_server_url>
```

**Note:** The `ACTIVATIONSERVER` option is only intended for first time activation. If you have previously signed-in to this version of Tableau, you cannot specify the activation server using this option. For example, if you are using Tableau Desktop version 2019.3 and have previously signed-in successfully, you cannot use the `ACTIVATIONSERVER` option to specify an activation server. However, if you install a different Tableau version or you are signing-in to Tableau for the first time on a new computer, you can use the `ACTIVATIONSERVER` option to specify the activation server.

Enable login-based license management on Tableau Desktop by editing the registry

If Tableau Desktop is already installed, you can still enable login-based license
management by editing the Windows registry to add an entry for LBLM.

```
reg.exe HKEY_LOCAL_MACHINE\SOFTWARE\Tableau\ATR /f /v LBLM /d enabled
reg.exe HKEY_LOCAL_MACHINE\SOFTWARE\Tableau\ATR /f /v LBLM /d disabled
reg.exe HKEY_LOCAL_MACHINE\SOFTWARE\Tableau\ATR /f /v LBLM /d required
```

1. As an administrator on the computer running Tableau Desktop, make a backup of the registry file before you make any changes to it.

2. Edit the registry, and in HKEY_LOCAL_MACHINE\SOFTWARE\Tableau, create a new hive named ATR with these values:
   a. Name: Add a string value named LBLM.
   b. Data: enabled.

3. Restart Tableau so the changes take effect.

To use a duration length other than the default of 14 days, include the `ATRRequestedDurationSeconds` registry setting. For example:

```
1. As an administrator on the computer running Tableau Desktop, make a backup of the registry file before you make any changes to it.
2. Edit the registry, and in HKEY_LOCAL_MACHINE\SOFTWARE\Tableau, create a new hive named ATR with these values:
   a. Name: Add a string value named ATREnabled.
   b. Data: Add 1.
   c. Name: Add a string value named ATRRequestedDurationSeconds.
   d. Data: Add a number of seconds the duration should last. For example, add 43200 to set a duration of 12 hours.
3. Restart Tableau so that the changes take effect.

Enable login-based license management on Tableau
Desktop on macOS

To enable login-based license management on macOS, run the following commands in a terminal window to update the preferences file, and then install or restart Tableau Desktop:

```bash
sudo defaults write /Library/Preferences/com.tableau.ATR LBLM "enabled"
sudo defaults write /Library/Preferences/com.tableau.ATR LBLM "disabled"
sudo defaults write /Library/Preferences/com.tableau.ATR LBLM "required"
```

To use a duration length other than the default of 14 days, include the `ATRRequestedDurationSeconds` preferences setting. For example:

```bash
sudo defaults write /Library/Preferences/com.tableau.ATR ATRRequestedDurationSeconds -string "43200"
```

**Note:** Login-based license management uses the following hierarchy when determining ATR duration.

1. **ATR Service** – Establishes the minimum (4 hours) and maximum (3 months) ATR durations applicable to all users/installations. It specifies the default ATR duration (2 weeks) if nothing is specified by Tableau Server or Tableau Desktop.

2. **Tableau Server** - Can optionally specify a maximum ATR duration (`licensing.login_based_license_management.max_requested_duration_seconds`) for all Tableau Desktop installations. This ATR duration setting overrides the ATR Service maximum setting (3 months) and must be within the ATR Service minimum (4 hours) and maximum (3 months).

3. **Tableau Desktop** – Can optionally specify the ATR duration (`ATRREQUESTEDDURATIONSECONDS`) for the computer on which it is installed. This setting must be within the ATR Service minimum (4 hours) and maximum (3 months).
This local ATR duration overrides any durations set by the ATR Service or Tableau Server. However, this default ATR duration cannot be more than the maximum ATR duration set on Tableau Server.

<table>
<thead>
<tr>
<th>Setting</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RECLAIMLICENSE</td>
<td>1</td>
<td>Deactivates any Tableau Desktop product key previously activated on the computer. When you reclaim a license, it makes Tableau Desktop unlicensed, so that users are prompted to activate their license after launching Tableau Desktop.</td>
</tr>
<tr>
<td>LBLM</td>
<td>disabled</td>
<td>The default. The login-based license management feature will not appear on the licensing screens.</td>
</tr>
<tr>
<td>LBLM</td>
<td>enabled</td>
<td>The login-based license management is enabled. The licensing screens will present the two options for activation (product key, or credentials).</td>
</tr>
<tr>
<td>LBLM</td>
<td>required</td>
<td>Login-based license management is the only way to activate the Tableau Desktop (when the licensing screen appears, it will offer only the credentials option for activation).</td>
</tr>
</tbody>
</table>

**Note:** When login-based license management is set to enabled or required, reporting is also enabled.

Additional configuration for Citrix Virtual Apps and Desktops users

If you are using login-based license management for Tableau Desktop on Citrix Virtual Apps and Desktops (formerly XenApp and XenDesktop), you may get an error message that your
license information has changed whenever you launch Tableau Desktop. This error forces a restart, which then asks you to register Tableau Desktop again. This error occurs because the ATR service sends a new token that doesn’t match the license cache.

To prevent this from occurring, you can use the SYNCHRONOUSLICENSECHECK and SILENTLYREGISTERUSER options with the Windows installer. For example:

tableau-setup-std-tableau-2019-4.19.1018.2338-x64.exe
SYNCHRONOUSLICENSECHECK="true" SILENTLYREGISTERUSER="true"

Or, you can edit the following registry keys:

Reg key path: HKLM\SOFTWARE\Tableau\<Tableau version>\Settings\

Reg key (String value, need to set to true to make that feature enabled)

SynchronousLicenseCheck

SilentlyRegisterUser

This error is believed to happen on macOS, but there isn’t a workaround at this time.

Note: These registry keys are not final and might change in future releases.

Step 3: Activate Tableau Desktop

The 2019.4 version of both Tableau Desktop for Windows or macOS support login-based license management.

1. Run Tableau Desktop setup.
2. The Activate Tableau screen will include the Activate with your credentials option.
3. Click **Activate with your credentials** and then do one of the following:
   - If you’re using Tableau Online, click the Tableau Online link.
   - If you’re using Tableau Server, enter the Tableau Server URL, and then click **Connect**.

   When prompted, enter valid credentials for a user with a Creator role subscription, and then click **OK**.

**View login-based license usage**

You can view login-based license usage for Tableau Online or Tableau Server. The report shows users, hosts, user role, product, version, activations, Creator seats in use, Creator seats not in use, and when a Creator seat was last used. You can view data for the past 30 days up to a maximum of 183 days.

To view the Login-based license management License Usage administrative view:
1. In Tableau Online or Tableau Server, in the navigation pane, click **Site Status**.

2. On the Site Status page, click **Login-based License Usage**.

3. Optional. On the report screen, you can change the time window to show when seats were last used, filter on actions, filter on user name, and sort by columns.
Troubleshooting

You may encounter one of the following error messages while using login-based license management. Use the information below to resolve the issue.

Login-based license management is not enabled on Tableau Server

If login-based license management is enabled on Tableau Desktop, but not enabled on Tableau Server, or if Tableau Server does not have a login-based license management capable license, the following error message may be displayed:

*Login-based license management is not enabled. Choose another activation option or contact your administrator to enable login-based license management.*

Contact your administrator to activate login-based license management on Tableau Server and enable the login-based license management configuration setting. In addition, ensure that you’re using a Tableau Server product key that’s enabled for login-based license management. Login-based license management should be enabled by default unless it was previously turned off by another user.
Login-based license management is not enabled on Tableau Desktop

If login-based license management is not enabled on Tableau Desktop, you will not have the option to activate Tableau using your credentials.

On the computer where Tableau Desktop is installed, enable login-based license management. For more information, see Step 2: Enable login-based license management.

You do not have a Creator license

When attempting to activate from Tableau Desktop, the following error message may be displayed:

*You do not have a Creator license. Contact your administrator to obtain one.*

This error is displayed if you have not been assigned a Creator role. Existing role-based product keys issued prior to 2019.2 do not have this capability. Verify that you have activated the appropriate product key. For guidance on activating licenses, see tsm licenses activate.

To verify that the product key you have activated on the server includes Creator licenses, navigate to:

http://<tableau_server>/#/server/licenses

You have activated the maximum number of computers

When attempting to activate from Tableau Desktop, the following error message may be displayed:

*You have activated Tableau the maximum number of times allowed under your account. You must wait for the license activation on another computer to expire before you can activate Tableau again.*

This error is displayed when you activated Tableau from multiple computers with the same Creator user credentials and exceeded the maximum number of activations. You must wait until the authorization-to-run (ATR) token expires on one of the existing computers before
Tableau Server on Linux Administrator Guide

attempting to activate a new computer. If you are using non-persistent virtual machines (VMs), you can shorten the ATR duration to prevent this error from occurring again.

To shorten the ATR token duration and free-up a user seat

If you encounter this error, you can shorten the ATR duration to 4 hours (14400 seconds) to free-up a seat.

1. Open a Command Prompt as an administrator.

2. Navigate to the Tableau binaries (`\bin`) directory, using the following command:

   `cd Program Files\Tableau\Tableau <version>\bin`

3. Run the following command to set the duration to 4 hours, in seconds (e.g., 14400):

   `atrdiag.exe -setDuration 14400`

4. Next, overwrite the existing ATR token. In Tableau Desktop, select **Help > Manage Product Keys**.

5. In the **Manage Product Keys** dialog box, reactivate Tableau Desktop through Tableau Server using login-based license management (which will overwrite the existing token).

6. Close Tableau Desktop and wait for the ATR duration to elapse (e.g., 4 hours) so that the ATR token expires and frees-up a user seat.

To return your computer to an unlicensed state

1. Open a Command Prompt as an administrator.

2. Navigate to the Tableau binaries (`\bin`) directory, using the following command:

   `cd Program Files\Tableau\Tableau <version>\bin`

3. Run the following command:

   `atrdiag.exe -deleteAllATRs`
Note: This removes only the ATR token from the computer. It does not free-up any of the user seats. The user seat is only freed-up after the deleted ATR token expires.

Your Tableau credentials are invalid

When attempting to activate from Tableau Desktop, the following error message may be displayed:

*Your Tableau credentials are invalid. Contact your administrator to reset your account.*

This error is displayed when you entered incorrect credentials for the account you are using to activate Tableau.

Your computer’s clock is not synchronized to the current time

When attempting to activate from Tableau Desktop, the following error message may be displayed:

*The Tableau license server at https://atr.licensing.tableau.com:443/ returned an error. If this issue persists, contact Tableau Support.*

This error is displayed when your computer’s clock is not synchronized with the current time. Synchronize your computer’s clock with a time server on the internet or enable automatic time synchronization.

Unable to activate with your credentials

When attempting to activate Tableau, the following error message may be displayed:

*Tableau Server cannot verify your licensing information over the internet. Contact your administrator to check your internet connection.*

This error is displayed when the port atr.licensing.tableau.com:443 is not open on all Application Server (VizPortal) nodes.
Add Users to Tableau Server

You can add users to Tableau Server one at a time or in batches. You can add them to the server as unlicensed users, and then add them to sites and assign site roles as you onboard them to Tableau Server. Or you can add users to sites and specify their site roles at the same time, at which point they are ready to sign in.

The steps below describe how to add an individual user and assign their site role. To add users in batches, you import them, providing Tableau Server with a CSV file that contains rows of users and their attributes. For information, see Import Users and CSV Import File Guidelines.

Considerations for Active Directory authentication

If your Tableau Server is configured for Active Directory authentication, before you add users, review User Management in Active Directory Deployments to understand how multiple domains, domain naming, NetBIOS, and Active Directory user name format influence Tableau user management.

**Note:** If you run Tableau Server on Linux, all external directory communication is configured and managed with a LDAP identity store. In the context of user and group synchronization, Tableau Server configured with LDAP identity store is equivalent to Active Directory. Active Directory synchronization features in Tableau Server function seamlessly with properly configured LDAP directory solutions.

Adding users at the server level vs. the site level

In a single-site environment, server administrators can add users on the Users page.
After you add a site to Tableau Server, it becomes a multi-site server with a **Server Users** page (all server users from every site appear here) and a **Site Users** page. Only server administrators can access the **Server Users** page, and both site administrators and server administrators can access the **Site Users** page.

The **Server Users** page is the only place where you can assign users to multiple sites, delete users from the server, and if the server is using local authentication, reset user passwords.
Add a user to the server

1. In the site menu, select **Manage All Sites > Users**, and then click **Add Users**.

   To add a user to a site, you select the site and go to the **Users** page.

2. Do one of the following:
   - If the server is configured for **local authentication**, click **New User**, and enter a user name. With local authentication, the best way to avoid user name collisions is to provide an email address for the user name. For example, `jsmith@example.com` instead of `jsmith`. 
User names are not case sensitive. Characters not allowed in user names include the semi-colon (;) and colon (:).

- If the server is configured for **Active Directory authentication**, click **Active Directory User**. If you are adding a user from the same Active Directory domain that Tableau Server runs on, the server domain will be assumed, and you can type the AD user name without the domain.

  **Note:** Do not enter the user’s full name; this can cause errors during the importing process.

3. If the server is using local authentication, provide the following:

   - **Display Name**—Type a display name for the user (e.g., *John Smith*).

   - **Password**—Type a password for the user.

   - **Confirm password**—Retype the password.

   - **Email**—This is optional and can be added at a later time in the user profile settings.

   - **Selected users are Server Administrators**: Specify whether the user should be a server administrator.

   - **Name (Site Membership) / Site Role**: If the user is not a server administrator, you can assign a user to zero or more sites, along with a site role for each site. You do not have to choose site membership and site role at this time. If you don’t specify site membership and site role for a new server user, the user will be added as a server user only, with a site role of Unlicensed. For details, see Set Users’ Site Roles.

4. Click **Create**.
Sign in to the Tableau Server Admin Area

As a server administrator on Tableau Server, you can access admin settings to configure sites, users, projects, and to do other content-related tasks.

If you want to change server settings such as processor, caching, authentication, distributed deployment, and other related configurations, see Sign in to Tableau Services Manager Web UI.

If you are running Tableau Desktop and want to sign in to Tableau Server to publish or access content and data sources, see Sign in to Tableau Server in Tableau Desktop.

Here’s how to sign in to the Tableau Server admin pages:
1. Open your browser and enter the server URL. Here are some examples of what the URL might look like:

`http://localhost/` (if you're working directly on the server computer)

`http://MarketingServer/` (if you know the server's name)

`http://10.0.0.2/` (if you know the server's IP address)

If the server is not using port 80, you need to include the port number in the URL, as in these examples:

`http://localhost:8000/`

`http://MarketingServer:8080/`

`http://10.0.0.2:8888/`

... where 8000 or 8080 or 8888 is the port that you configured.

Tableau Server displays a page where you can enter a user name and password:
2. Enter the credentials for the server administrator that you created when you finished installing Tableau Server.

You’re then taken to the main page of the Default site, and you’re ready to create users, sites, and manage content.

Navigate the Admin Areas of the Tableau Web Environment

As an administrator on Tableau Server or Tableau Online, you can access admin settings that aren’t available to other users to configure sites, users, projects, and to do other content-related tasks.

The settings in this article refer to the Tableau web environment. Tableau Server administrators with appropriate credentials can also change server settings such as processor, caching, authentication, distributed deployment, and related configurations using the TSM web environment. For information, see Sign in to Tableau Services Manager Web UI.

Access based on site role and number of sites

The menus you get when you sign in to Tableau Server or Tableau Online depend on the following conditions:

- Whether you’re a site or server administrator.

  Site administrator access is available on Tableau Online and Tableau Server. Server administrator access is only on Tableau Server.

- Whether you have access to only one site or to multiple sites.
Server administrator

On a **single-site** server, the site selector does not appear, and all other menus are the same.

In a **multi-site** environment, menus along the left enable you to modify a specific site or all sites, and to configure users, groups, schedules, tasks, and server settings.

To access server administrator settings that affect all sites, open the site menu by clicking the arrow next to the current site name, and then select **Manage all sites**.

The **Content** and **Group** tabs go away, and the site menu text changes to **All Sites** to let you know you are managing server-wide settings, and options like **Server Status** reflect the server-wide view.
To return to the site administration menus, select **All Sites**, and then select the site you want to manage.
Site administrator

If you are a site administrator for Tableau Online or Tableau Server, and you have access to multiple sites, you’ll get menus for selecting which site to manage, and for managing that site’s content, users, groups, schedules, and tasks, and for monitoring its status.

The site selector displays the name of the current site. To go to another site, select the site menu, and then select the site name.

If you have access to only one site, the site selector does not appear, but all other menus are the same.

Server administrator tasks

Server administrators (available with Tableau Server Enterprise only) can do the following:

- Monitor server status and activity.
- Generate log files.
- Add sites and edit site settings. Only server administrators can add sites to the server.
- Add users to the server, and assign users to sites.
- Add and manage site groups.

To manage settings only for a specific site, you must first navigate to the site. Within each site, you can do the following:
Tableau Server on Linux Administrator Guide

- Administer content: Create projects, move content from one project to another, assign permissions, change ownership of a content resource, and so on.

- Manage schedules for extract refreshes and subscriptions.

- Monitor site activity and record workbook performance metrics.

- Manage storage space limits for content published by users.

- Allow web authoring.

- Enable revision history.

- Allow site administrators to add and remove users.

- Allow users to subscribe to workbooks and views, and allow content owners to subscribe others to workbooks and views.

- Enable offline snapshots for favorites (iOS only).

Site administrator tasks

A site administrator on Tableau Online or Tableau Server can do the following tasks:

- Administer content: Create projects, move content from one project to another, assign permissions, change ownership of a content resource, and so on.

- View, manage, and manually run schedules for extract refreshes and subscriptions.

- Add and manage site users (if allowed by the server administrator; see Add or Update Sites).

- Add and manage site groups.

- Monitor site activity.
Sign in to Tableau Services Manager Web UI

This topic explains how to sign in to the Tableau Services Manager (TSM) web UI. The TSM web pages are used to configure Tableau Server settings such as user authentication, server processes, caching, and other server-related settings. You can also configure TSM from a command line shell. See tsm Command Line Reference.

**Important:** Signing into TSM is *not* the same as signing into Tableau Server. TSM is used to configure the server, and requires an account with administrative privileges on the computer running TSM. Depending on how your enterprise is organized, the TSM administrator could be a user who does not have a Tableau Server account. A Tableau Server administrator has access to administrative pages for creating and editing sites, user, product, and other content-related tasks. For information about signing into Tableau Server as a Tableau Server administrator, see Sign in to the Tableau Server Admin Area.

If you are running Tableau Desktop and want to sign in to Tableau Server to publish or access content and data sources, see Sign in to Tableau Server in Desktop.

**Requirements**

- The account that you use to sign in to TSM must have administrative access to the local computer where Tableau Server is installed.

  Specifically, the account that you use to sign in to TSM must be a member of the tsmadmin group that was created during initialization. To view the user accounts in the tsmadmin group, run the following command in the Bash shell:

  ```bash
grep tsmadmin /etc/group
  ```

  If the user account is not in the group, run the following command to add the user to the tsmadmin group:

  ```bash
sudo usermod -G tsmadmin -a <username>
  ```
Tableau Server on Linux Administrator Guide

Tableau Server on Linux relies on PAM for core authentication scenarios. For more information about PAM integration for TSM administration, see TSM Authentication.

- If you are running a distributed deployment of Tableau Server, then enter the host name, or IP address of the computer running the initial node. The credentials you enter must have administrative access to the computer running the initial node, as specified above.

- Specify an HTTPS protocol in the server URL. Tableau Server installs a self-signed certificate as part of the installation process. Therefore, the protocol must be specified as https. For more information about the self-signed certificate and certificate trust for TSM connections, see Connecting TSM clients.

- Specify the port for TSM web UI (8850) in the URL.

- If you are running a local firewall, open port 8850. See Configure Local Firewall.

- You must specify the hostname or IP address of the computer running TSM. If you have set up a load balancing or proxy solution in front of Tableau Server, do not specify the load balancer or proxy address.

Sign in to the TSM web UI

1. Open a browser and enter the Tableau Server URL, and append the dedicated TSM web UI port.

   Here are some examples of what the URL might look like:

   https://localhost:8850/ (if you're working directly on the server computer)

   https://MarketingServer:8850/ (if you know the server's name)

   https://10.0.0.2:8850/ (if you know the server's IP address)

2. In the sign-in page that appears, enter your administrator user name and password.
**Note:** Tableau Server creates and configures a self-signed certificate during the installation process. This certificate is used to encrypt traffic to the TSM Web UI. Because it's a self-signed certificate, your browser will not trust it by default. Therefore, your browser will display a warning about the trustworthiness of the certificate before allowing you to connect.

---

**Customize Your Server**

You can customize the Tableau Server web pages to personalize it for your company or group. You can perform these customizations:
Tableau Server on Linux Administrator Guide

- Change the server name that appears in the browser tab, tooltips, and messages.
- Change the logos that appear in the web environment.

For more information, see tsm customize.

- Set the language used for the web environment and the locale used for views. See Language and Locale for Tableau Server.
- Install custom fonts on Tableau Server and client computers that connect to Tableau Server. See Use Custom Fonts in Tableau Server.
- Add a custom note to the server sign in page. The Sign In setting lets you add text. You can optionally add a URL to make the text a link. The sign in page text does not display on Tableau Mobile.

To set a custom note, sign in to a site on Tableau Server. On the left-side navigation pane, select Manage all sites from the drop-down site list. Select Settings and add a message to Sign In Customization.

For more information, see Sign In.

- Add a custom message to the welcome banner on the home page for all server users to see. The custom message can contain up to 240 characters of text and hyperlinks as well as one paragraph break. Administrators can also disable the default Tableau welcome banner for the server.

To set a custom welcome banner, sign in to a site on Tableau Server. On the left-side navigation pane, select Manage all sites from the drop-down site list. Select Settings, then navigate to the Customization page.

Administrators and project leaders can also add images for projects in thumbnail view.

Language and Locale for Tableau Server

Tableau Server is localized into several languages. Server language and locale settings impact how this affects users. The Language setting controls user interface (UI) items such
as menus and messages. The **Locale** setting controls items in views such as number formatting and currency.

Administrators can configure language and locale on a server-wide basis and individual users can configure their own settings (search for "Your Account Settings" in the Tableau Server Help). If a user configures their own language and locale, their settings override the server settings.

### Supported Languages

Tableau Server is localized into several languages. See the "Internationalization" section of the [Tableau Server Technical Specification](#) page for more information.

### Default Settings

The default language for Tableau Server is determined during Setup. If the host computer is configured for a language Tableau Server supports, Tableau Server installs with that language as its default. If computer is configured for a language that is not supported, Tableau Server installs with English as its default language.

### How Language and Locale are Determined

Another influence on which language and locale display when a user clicks a view is the user’s web browser. If a server user has not specified a **Language** setting on their User Account page, and their web browser is set to a language that Tableau Server supports, the browser’s language will be used—even if Tableau Server itself is set to a different language.

Here’s an example: Assume that Tableau Server has a system-wide setting of English as the **Language** for all users. Server user Claude does not have a language specified on his Tableau Server User Account page. Claude’s browser uses German (Germany) for its language/locale.

When Claude signs in to Tableau Server, the server UI displays in German and when he clicks a view, the view uses the Germany locale for numbers and currency. If Claude had
set his user account Language and Locale to French (France), the UI and view would have been displayed in French. His user account setting supersedes those of his web browser, and both of those have precedence over the Tableau Server system-wide setting.

Another setting to be aware of is the Locale setting in Tableau Desktop (File > Workbook Locale). This setting determines the locale of the data in the view, such as which currency is listed or how numbers are formatted. By default, Locale in Tableau Desktop is set to Automatic. However, an author can override that by selecting a specific locale. Using the above example, if the author of View A set Locale to Greek (Greece), certain aspects of the data in View A would display using the Greek (Greece) locale.

Tableau Server uses these settings, in this order of precedence, to determine language and locale:

1. Workbook locale (set in Tableau Desktop)
2. Tableau Server User Account language/locale settings
3. Web browser language/locale
4. Tableau Server Maintenance page language/locale settings
5. Host computer’s language/locale settings

Use Custom Fonts in Tableau Server

You can use custom fonts with Tableau Server. When you do this the safest way to guarantee that users have the experience you intend is to keep the following in mind:

- The fonts need to be installed on the computer where Tableau Server is running. After installing the fonts, restart Tableau Server to use the new fonts.
- The fonts need to be installed on any client computers that will connect to Tableau Server. You need to have the fonts installed locally in order for your browser to
properly display them.

- As a best practice, use "web safe" fonts that are installed by default on all major browsers. This increases the likelihood that the fonts will display properly on client machines.

- Different browsers render the same fonts differently, so even when a client browser has the custom font installed, it may look different when viewed in different browsers. This can be especially noticeable with comments or titles where specific spacing is used for an intentional effect.

**Note:** For more information about installing fonts on Linux, refer to your Linux distribution's documentation and support.

Manage Sites

You can plan and manage your sites in Tableau Server. You can manage users and groups for your sites, manage projects and content access, manage data, and create and interact with views on the web.

Sites Overview

The topics in this section describe the Tableau Server concept of a site and aspects of working with multiple sites. Topics include authentication type each site uses, and what to know about user licenses and administrator-level access to sites.

What is a site

You might be used to using the term _site_ to mean “a collection of connected computers,” or perhaps as the short form of “website.” In Tableau-speak, we use _site_ to mean a collection of users, groups, and content (workbooks, data sources) that’s walled off from any other groups and content on the same instance of Tableau Server. Another way to say this is that
Tableau Server supports multi-tenancy by allowing server administrators to create sites on the server for multiple sets of users and content.

All server content is published, accessed, and managed on a per-site basis. Each site has its own URL and its own set of users (although each server user can be added to multiple sites). Each site’s content (projects, workbooks, and data sources) is completely segregated from content on other sites.

For site administrator recommendations for how to set up users on a site, how to structure a site for publishers and other content users, how to give users permissions to share and manage their content, and so on, see the Manage Sites section.

For information about how users can get their content to Tableau Server, see Publish Data Sources and Workbooks in the Tableau user help.

Authentication and sign-in credentials

By default, all sites on a server use the same identity store type. You configure these settings when you install Tableau Server. For information, Configure Initial Node Settings.

Users who have access to more than one site on the same Tableau Server instance use the same credentials for each site. For example, if Jane Smith has a user name of jsmith and a password of MyPassword on Site A, she uses those same credentials on Site B. When she signs in to Tableau Server, she’ll be able to choose which site she wants to access.

The Default site

Tableau Server installs with a site named Default. If you maintain a single-site environment on Tableau Server, this becomes the site you work with, and on which your users share their Tableau analysis. If you add sites, Default becomes one of the sites you can select when you sign in to Tableau Server. Default differs from sites that you add to the system in the following ways:

- It can never be deleted but, just like sites that you add, it can be renamed.
- It stores the samples and data connections that ship with Tableau Server.
The URL used for Default does not specify a site. For example, the URL for a view named Profits on a site named Sales is http://localhost/#/site/sales/views/profits. The URL for this same view on the Default site would be http://localhost/#/views/profits.

Why or why not add sites

On Tableau Server, users, projects, groups, data sources, and workbooks are managed per site. You can add users to multiple sites.

Each environment and its needs is unique. However, as a baseline, Tableau Zen Masters and Product Managers tend to recommend using sites for true multi-tenancy needs. In other words, create a new site only when you need to manage a unique set of users and their content completely separately from all other Tableau users and content.

For site administrator recommendations for how to set up users on a site, how to structure a site for publishers and other content users, how to give users permissions to share and manage their content, and so on, see the Manage Sites section.

Examples for which it makes sense to use sites

- You are a consultant who manages Tableau analysis for multiple clients, and you want to create a site for each client, to ensure that data from one client is not exposed to another.
- You want to allow Guest user access to a small and contained area of the server.

Examples for which projects can work better than sites

- A content-development process in which data sources and reports evolve from sandbox to production phases.

Migrating users and content from one site to another is a laborious process. Although you might have good reasons to use sites for this and similar processes, by creating sites, you as the site administrator compound your ongoing maintenance burden. For each configuration update you make to one site (for example creating new projects
and setting permissions), you usually would need to duplicate the same work on each additional site.

- You want to separate areas of the server by functional area.

Among a group of Tableau users, it’s common that some users need to access content in multiple areas. Using sites would encourage publishing the same data sources and reports to multiple sites. This leads to data source proliferation and can negatively impact server performance. Using projects is a simpler way to work with this scenario.

For additional ideas, see the following resources:

- Why use projects in the topic User Projects to Manage Content Access.

- Discussions about sites on the Tableau Community forums. Here's a link to get you started.

**Administrator-level access to sites**

Tableau Server includes three administrator-level site roles: Server Administrator, Site Administrator Creator, and Site Administrator Explorer.

The **Server Administrator** site role always takes the highest license available, and it allows full access to Tableau Server, including all content access. You can find more information about this role in Server Administrator Overview. Server administrators also create sites as needed. (Site administrators don’t have permissions to do this.)

A server administrator can assign one of the **Site Administrator** site roles to users to delegate creating and maintaining a specific site’s user and content framework. The content framework enables Tableau users to share, manage, and connect to data sources and workbooks.

- Assign **Site Administrator Creator** to administrators who also connect to data, and create and publish data sources or workbooks. This site role takes a Creator license.

- Assign **Site Administrator Explorer** if the user manages the content framework but
doesn’t need to edit the content itself. This site role takes an Explorer license, and it allows viewing and interacting access.

By default, the Site Administrator site roles allow creating and managing the site’s users and groups, creating projects to organize content on the site, assigning permissions to allow users (groups) to access the content they need, scheduling extract refreshes, and a few other tasks.

A server administrator can deny site administrators’ user management tasks. For example, you might do this if you use the Site Administrator Creator role for the data experts. In other words, you want to allow these users to manage connections to underlying data, create and publish “single source of truth” data sources, create top-level projects, and organize content across projects without restriction; but not necessarily add and remove site users.

For each site the server administrator can also limit site administrator access, so that site administrators can manage groups and content, but not add or remove users or set users’ site roles.

In some organizations, the same person might be both a server administrator and site administrator for one or more sites. Even so, the tasks performed by a site administrator and a server administrator are distinct.

Licensing and user limits

You can add server users to multiple sites, and set their site roles and permissions on each site. A user who belongs to several sites does not need a license for each site. Each server user needs only one license.

Server administrators can use the Limit number of users setting (select Site <name> > Settings) to specify a user limit for the site. Only licensed users are counted. For example, if a site has 90 licensed users (including administrators), 20 unlicensed users, the user count is 90.
Add or Update Sites

Tableau Server comes with one site named Default. Server administrators can add sites to the server and modify any site’s settings.

Add a site

1. Do one of the following:

   - If you’re adding a site to the server for the first time, select Settings > Add a Site, and then click Add a Site.

   ![Add a Site](image)

   - If you’ve added sites before, in the site menu, click Manage All Sites, and then click New Site.

   ![Manage All Sites](image)

   - If you’re editing an existing site, select it on the Sites page, and then select Edit Settings. Or, in a single-site deployment, click Settings, and then click the Gen-

   ![Edit Settings](image)
2. Enter a **Site Name and ID**. (If you are editing the Default site, you cannot change the ID.)

   **Note** The “#/site” portion of the URL (for example, http://localhost/#site/sales) cannot be changed. In multi-site server environments, these segments appear in the URL for sites other than the Default site.

3. For **Storage**, select either **Server Limit** or **GB**, and enter the number of GB you want as a limit for storage space for published workbooks, extracts, and other data sources.

   If you set a server limit and the site exceeds it, publishers will be prevented from uploading new content until the site is under the limit again. Server administrators can track where the site is relative to its limit using the **Max Storage** and **Storage Used** columns on the Sites page.
4. Under Manage Users, select whether only server administrators can add and remove users and change their site roles, or whether you site administrators can too.

   If you are allowing site administrators to manage users, specify how many users they can add to the site by selecting one of the following:

   - **Server limit**: For a server with user-based licensing, the limit is the number of available server seat licenses. For a server with core-based licensing, there is no limit. For more information, see View Server Licenses and Troubleshoot Licensing.
   
   - `<number of> users`: Allows a site administrator to add users up to a limit you specify. See Work with Sites for information on licensing and user limits.

5. Under **Web authoring**, select whether browser-based authoring is enabled for the site.

   When web authoring is disabled, users cannot edit published workbooks from the server web environment. To update a workbook published to the server, a Tableau Desktop user must re-publish it. For more information, see Set a Site's Web Authoring Access and Functions.

6. Enable **Comments** so users can share a conversation about data views, including snapshot images to highlight discoveries. For more information, see Comment on Views in Tableau user Help.

7. Enable **Data-driven alerts** to let users automatically receive emails when data reaches key thresholds. For more information, see Send Data-Driven Alerts in
8. Under **Subscriptions**, you can let site users subscribe to views and receive regular emails of them. Select **Let content owners subscribe other users** to let administrators, project leaders, and content owners set up subscriptions for others. These options are visible only if you have configured subscription settings.

9. Under **Email Settings**, you can enter a custom **From Address** for alerts and subscriptions. While the email address you enter should use valid syntax (such as bizdev@myco.com or noreply@sales), Tableau Server does not require it to correspond to a real email account. (Some SMTP servers may require an actual address, however).

   If you want a standard email footer to appear, select **Custom footer**, and enter the text you want to display above Tableau Server links in email messages.

   The email footer will look similar to the following:

10. **Enable offline favorites** for Tableau Mobile to let mobile users access cached previews of views.

11. Select **Record workbook performance metrics** to permit your site users to collect
metrics on how workbooks perform, such as how quickly they load

To initiate recording, users must add a parameter to the workbook’s URL. For more information, see Create a Performance Recording.

12. Click Create or Save.

**Note:** After you add your first site to Tableau Server, the site menu becomes available. When you select All Sites and click Users, the Server Users page appears, showing all users. When you select a specific site and click Users, the Site Users page appears. To add users to the server or specific sites, see Manage Users and Groups.

Export or Import a Site

You can provision a new Tableau Server site by importing (migrating) information from another site. You do this by exporting the existing site’s (the source site) information to a file. Then you complete steps to verify and import that information to the target site.

Site migration options

You can migrate a site in any of these ways:

- To another site on the same Tableau Server instance.
- To a site on a separate Tableau Server instance.
- From Tableau Online to your Tableau Server deployment.

Many of the tips listed in Prepare the source and target sites apply to the Tableau Online-to-Tableau Server migration. For more information about the process for migrating from Tableau Online to Tableau Server, see Migrate from Tableau Online to Tableau Server. In addition, you should become familiar with What information is or isn’t preserved in a site export.
Note: After you prepare your Tableau Online site for export, contact your Tableau representative. Tableau will complete the export steps, and send you the export files, which you can then import to your site on Tableau Server.

- From Tableau Server on Windows to Tableau Server on Linux or vice-versa (or cross-platform Tableau Online-to-Tableau Server site migration).

What information is or isn’t preserved in a site export

The export file you create preserves workbooks, projects, data sources, and users. This includes permissions set on content, user favorites lists, and site quotas.

Users’ custom views are preserved; however, depending on the type of site migration, custom view URLs might change in a way that breaks users’ bookmarks to their views. For example, migrating from Tableau Online to a self-managed Tableau Server will change the host name and invalidate bookmarks.

In addition:

- When you export a site on Tableau Server to import to another Tableau Server site, subscription and extract refreshes schedules are preserved.

- When you export a Tableau Online site to import to Tableau Server, only the default schedules are preserved.

  On Tableau Online, users can create their own custom subscription schedules. These schedules currently are only supported on Tableau Server version 2018.2 if the administrator has followed the steps provided in Enable Custom Schedules for Subscriptions.

Usage data, which appears in the site’s administrative views, is not preserved. For example, view and data source counts, user actions, and performance data.
Backgrounder jobs that are in-progress while a site is being exported, will not be exported and will not show up on the new site once the import is complete.

In addition, OAuth access tokens embedded in data connections are reset. For those data sources, you will need to edit the connections and re-authenticate to the underlying data.

Prepare the source and target sites

Before you export a site, complete the following checklist to prepare both environments. Some of these instructions depend on whether both sites are on the same server instance or on separate ones.

• **Verify version compatibility**
To import a site successfully, the source and target sites must be in the same product version family. For example, 2019.4 through 2019.4.x. You cannot import a site that is outside the target site’s version family. For example, 2019.3 to 2019.4. For version information, see the Tableau Release Notes.

• **Identify node**
The `tsm sites` commands will use your local file store to hold the export and import data. If you are running a multinode Tableau cluster, then you must run the `tsm sites` commands on a Tableau Server that is running the Data Engine process. For information about the Data Engine process and the processes that require it, see Tableau Server Processes.

• **Delete stale content items**
Make sure the source site contains only what you want to import to the new site. Delete unused workbooks, data sources, or projects.

• **Remove obsolete users**
Confirm that all server users are licensed, and remove accounts that are no longer in use. You can’t remove users during the import process, so if the two sites are on the same server instance, all users you export from the source site are imported to the target site.

• **Create or identify the target site**
You must import a site file to a site that already exists on the target Tableau Server instance.
Because the import process removes anything from the target site that is not included in the import file, we recommend that you import to an empty site. For more information, see Add or Edit Sites.

Notes

- If the target site is not empty, workbooks and data sources with names that are identical on both target and source sites are replaced. This can be verified by the time stamp.

- If your source site has workbooks that use published data sources, the data connections in these workbooks continue to refer to the source site name. To maintain those connections without having to republish the workbooks, make sure the target and source site names match.

• Locate site IDs
The tsm command you use to export or import a site requires a parameter that takes the site ID. You can get the site ID from the URL when you’re signed in to the site from a web browser.

If only one site exists on the server, that site is named Default. When you’re signed in to the Default site, the browser URL looks something like this:

https://server-name/#/projects

In the URL, the absence of the /site parameter indicates that it’s the Default site. The site ID for the Default site is "Default" without the quotation marks.

On a multi-site Tableau Server deployment, the browser URL includes #/site/ followed by the site ID. The following URL would appear if you navigate to the Views page on a site whose site ID is finance:

https://localhost/#/site/finance/views
• **Check the identity store**
  You can export from and import to sites that do not use the same user identity store type, but you will need to modify the mapping files used for the import. This step is built into the import process and described in 3. Verify that site settings are mapped correctly.

• **Create users on the target server if necessary**
  The site import process assigns users to the target site. If the source site is on Tableau Online or on a Tableau Server instance other than the target site, you must create users on the target server before you can perform the import. If the two sites are on the same Tableau Server instance, the target site has access to the existing users, and you can skip this step.

User names are stored differently in Tableau Online, so when you export a Tableau Online site, you must edit the user-specific mapping file as part of the verification steps.

• **Configure the target server to deliver subscriptions**
  Subscriptions are imported, but you must configure the server to deliver them.

For more information, see Set Up a Site for Subscriptions.

• **Check schedules**
  The **Schedules** page lists the existing schedules for extract refreshes and subscriptions.

![Schedule Table](image)
For migrations from one Tableau Server site to another, refreshes and subscriptions assigned to default schedules on the source site are mapped to the same schedules on the target site.

If the source site has schedules that do not exist on the target site, and the target site is on another Tableau Server instance, you must create the schedules on the target site that you want the source schedules to map to. You can edit the mapping files to make sure this is done as you expect.

For migrations from Tableau Online to Tableau Server, custom subscription schedules that users create are not mapped. Before you contact Tableau to request your site export, remove these custom schedules.

To remove them, go to Tasks > Subscriptions, and select the check boxes for the custom subscriptions. Then, on the Actions menu, select Unsubscribe.

Tips for importing to a target with fewer users or schedules than the source site

When a target site has fewer users or schedules than the source site, many-to-one importing is not supported. How you can address this depends on whether the source and target sites are on the same Tableau Server instance.

Note: Subscription schedule information in this section does not apply to Tableau Online-to-Tableau Server site migration.
You can take any of the following approaches that apply to your situation:

- Remove extra users or schedules from the source site before you export.
  This is the preferred option if the two sites are on the same server instance.

- Add missing users or schedules to the target site before beginning the import.
  This is required if the target site is on another server instance.

- Add the missing users or schedules to the target site in the middle of the import process and manually update the mapping files.
  This is an option only if the sites are on the same instance.

- Manually map the users or schedules to different users and schedules in the target site during the import process.
  This is required if a user name differs between servers—for example, the exported user named adavis@company.com is defined on the target site as davisa.

1. Export a site

On the Tableau Server machine, type the following command:

```bash
tsm sites export --site-id <source-siteID> --file <filename>
```

Tableau Server must be running when you use the `export` command. During the export process, Tableau Server locks the site you’re exporting.

For example, to export a site with site ID `weather-data` to the file `export-file.zip`, type the following:

```bash
tsm sites export --site-id weather-data --file export-file
```

By default, Tableau Server saves `<export-file>.zip` to `/var/opt/tableau/tableau_server/data/tabsvc/files/siteexports`. For more information, see `tsm File Paths`.
2. Generate the import mapping files

To generate import files for the target site, you need the .zip file you created when you completed the steps in 1. Export a site.

1. On the Tableau Server machine, copy the exported .zip file to the directory Tableau Server expects to find the files for importing. For example:

   /var/opt/tableau/tableau_server-
   /data/tabsvc/files/siteimports

2. Verify that the target site already exists on Tableau Server. The import process will not create a site. See the previous section, Prepare the source and target sites.

3. Run the following command (Tableau Server must be running):

   tsm sites import --site-id <target-siteID> --file <export-file.zip>

   This command generates a set of .csv files that show how source site settings will map to the target site. In the steps described in the next section of this article, you confirm these mappings and adjust them where needed.

   By default these .csv files are generated to a mappings directory created under siteimports. For example:

   /var/opt/tableau/tableau_server-
   /data/tabsvc/files/siteimports/working/import_<id>_<date-time>/mappings

   For more information, see tsm File Paths.

3. Verify that site settings are mapped correctly

The .csv files you generated in the previous section describe how the source site’s resources will be assigned to the target site when the import is complete. Items in the files that Tableau Server was unable to map, and that you need to edit, are indicated by a series
of question marks (???). Before you can complete the import process, you must replace the question marks with valid assignments on the target site.

**Important:** Some requirements apply to mapping users, schedules, and published content resources, particularly when the source and target sites are on separate Tableau Server instances. For more information, see Prepare the source and target sites earlier in this article.

To verify mapping files

1. Navigate to the directory that contains the .csv map files generated by the `tsm sites import` command. By default:

   `/var/opt/tableau/tableau_server-/data/tabsvc/files/siteimports/working/import_<id>_<date-time>/mappings`

2. Use your preferred text editor to open one of the .csv files in the `mappings` directory, and do the following.

   a. Confirm that the mappings are correct.

   b. If an entry shows a series of question marks (???), replace them with a valid value.

      For descriptions of the settings in each of these files, use the tables in Mapping file content reference later in this article.

   c. Save the changes and preserve the CSV file formatting.

Repeat this process for the remaining .csv files.
4. Import the correctly mapped files to the target site

After you verify the site mappings in the .csv files, you can import the settings to the new site to complete the migration process.

1. Run the following command:

   ```
tsm sites import-verified --import-job-dir <import-id-directory> --site-id <target-siteID>
   ```

   For example:

   ```
tsm sites import-verified --import-job-dir /var/-
   opt/tableau/tableau_server-
   /data/tabsvc/files/siteimports/working/import_ff00_20180102022014457
   --site-id new-site
   ```

2. When the success message appears, sign in to the new site and confirm that everything was imported as you expected.

   **Note:** The `tsm sites import` and `tsm sites export` commands can leave a site in a locked state if an error occurs. To unlock a site, use the `tsm sites unlock` command.

Mapping file content reference

The following tables list the columns in each of the mapping files created when you run the `tsm site import` command.

CSV file name: mappingsDomainMapperForGroups

<table>
<thead>
<tr>
<th>Column title</th>
<th>Can it be edited?</th>
<th>Description</th>
</tr>
</thead>
</table>

1040 Version: 2019.4 Tableau Software
**source_name**  |  No  |  A user group name on the source site.

**source_domain_name**  |  No  |  The identity store type on the source site: either local (for local identity store) or a domain name (for Active Directory or LDAP external identity store).

**target_domain_name**  |  Yes*  |  The identity store type on the target site: either local for local identity store, or a domain name (such as example.com or example.lan) for Active Directory or LDAP external identity store.

*For the All Users group, keep the target_domain_name value set to local, even if your target server is configured for Active Directory identity store. The All Users group is a special default user group that must exist on every Tableau Server.

CSV file name: mappingsScheduleMapper

<table>
<thead>
<tr>
<th>Column title</th>
<th>Can it be edited?</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>source_name</td>
<td>No</td>
<td>The names of custom and default extract or subscription schedules on the source site.</td>
</tr>
<tr>
<td>source_scheduled_action_type</td>
<td>No</td>
<td>The type of schedule, either Refresh Extract, for extract refreshes, or Subscriptions, for subscription deliveries on the source site.</td>
</tr>
<tr>
<td>target_name</td>
<td>Yes</td>
<td>The names of custom schedules on the tar-</td>
</tr>
</tbody>
</table>
get site. You can edit this value. For example, if the schedule is named **Friday Update** on the source site you can rename it **Friday Refresh** on the target site.

<table>
<thead>
<tr>
<th>Column title</th>
<th>Can it be edited?</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>target_scheduled_action_type</td>
<td>No*</td>
<td>The type of schedule, either <strong>Refresh Extract</strong>, for extract refreshes, or <strong>Subscriptions</strong>, for subscription deliveries on the target site.</td>
</tr>
</tbody>
</table>

*In the rare case that you see question marks (???) in this column, replace them with either **Refresh Extract** or **Subscriptions**, to match the entry you see under **source_scheduled_action_type**.

CSV file name: mappingsSiteMapper

<table>
<thead>
<tr>
<th>Column title</th>
<th>Can it be edited?</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>source_url_namespace</td>
<td>No</td>
<td>The site ID of the source site.</td>
</tr>
<tr>
<td>target_url_namespace</td>
<td>No</td>
<td>The site ID of the target site.</td>
</tr>
</tbody>
</table>

CSV file name: mappingsSystemUserNameMapper

<table>
<thead>
<tr>
<th>Column title</th>
<th>Can it be edited?</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>source_name</td>
<td>No</td>
<td>The user name attribute of a user on the source site.</td>
</tr>
<tr>
<td>source_domain_name</td>
<td>No</td>
<td>The identity store type on the source site: either <strong>local</strong> (for local identity store) or a</td>
</tr>
</tbody>
</table>
| target_name | Yes | The user name attribute for users who will be assigned to the target site upon import. Confirm that all the user names in the list exist on the target server, and replace question marks (?) with user names that exist on the target server.

You cannot create user names by adding rows to the CSV file. Similarly, you cannot remove user names by deleting rows.

You can edit a user name in the target_name column to be different from its source user name, as long as the user already exists on the target server with that name.

For example, a user can have a source_name value of agarcia@company.com and a target_name value of ashleygarcia@company.com.

You can map a user on the source site to only one user name on the target site. |
| target_domain_name | Yes | The identity store type on the target site: either local (for local identity store) or a domain name (for Active Directory or LDAP external identity store). |
Delete Sites

Server administrators can delete sites that have been added to Tableau Server. Deleting a site also removes workbooks and data sources that were published to the site, as well as users. If a user belongs to additional sites, they will not be removed. To permanently delete a user, go to the Server Users page.

**Note:** The Default site cannot be deleted.

1. On the site menu, click **Manage all sites**, and then click **Sites**.
2. Select the site you want to remove, and then on the **Actions** menu, click **Delete**.
3. Click **Delete** in the confirmation dialog box that appears.

Site Availability

A site can become suspended or locked due to a site import failure, or because a server administrator chooses to suspend the site for a period of time.

When a site is suspended, only the server administrator can activate the site to make it available again.

**Note:** If a site becomes locked and you cannot access the Sites page through the Server interface, use the tsm sites unlock command to change the state to active.
To activate or suspend a site

1. In the site menu, click **Manage All Sites**, and then click **Sites**.

2. Select the site, and then select **Actions > Activate** or **Suspend**.

Allow Users to Save Revision History

Revision history enables your users to see how workbooks and data sources (content resources) have changed over time. Each time someone saves (publishes) a content resource, Tableau Server creates a new version, which becomes the current version. It makes the previous version the most recent revision in the revision history list. Revision history gives users confidence to experiment with their content, knowing that their older versions are available.

Notes

- This information applies to Tableau Server, and is for server administrators who want to allow publishers to work with revisions.

- On Tableau Online, workbook revision history (and not data source revision history) is
enabled on all sites. Users can save up to 10 revisions.

- For information about working with the content revisions themselves, including potential issues, see Work with Content Revisions in the User/Analyst section of the Tableau help.

Permissions users need to work with revision history

To access revision history, a user must have a site role of Creator or Explorer (Can Publish), plus the following permissions, depending on the content type:

- Project: View and Save
- Workbooks in the project: View, Save, and Download Workbook/Save As
- Data sources in the project: View, Save, and Download Data Source

Enable revision history and set the number of revisions allowed

Revision history is set at the site level, and is enabled by default, with a limit of 25 revisions for each content resource.

1. Sign in to a site as a Server Administrator, and click Settings.

2. Under Revision History, select Save a history of revisions, and enter the maximum number of revisions you want to allow for each content resource.

3. Click Save.

When you lower the number of revisions, the most recent revisions are saved. For example, if you set the limit to 15, the 15 most recent versions of the workbook or data source are saved.

**Note:** A content resource's revision history list might not show the changes to the limit until a background cleanup process runs on the server.
Clear all revisions

Server administrators can delete all previous revisions of published workbooks and data sources from a site. The most recent version of each published workbook and data source is always retained.

1. Sign in to a site as a Server Administrator, and click **Settings**.

2. Under **Revision History**, click **Clear Revision History**.

3. Click **Save**.

Security for previewing and restoring workbooks

When users select **Restore** or **Preview** for workbook revisions, user passwords are exchanged between the user’s browser and the server. Tableau Server encrypts these passwords using public/private key encryption. To ensure these public keys are provided by Tableau Server, you must configure the server to use SSL (HTTPS). For more information, see SSL.

See also

**Potential revision history issues** in the User/Analyst section of the Tableau help.

Extract Refresh Schedules

Tableau Desktop authors and data stewards can create and publish extracts. Extracts are copies or subsets of the original data. Because extracts are imported into the data engine, workbooks that connect to extracts generally perform faster than those that connect to live data. Extracts can also increase functionality.

Before refreshing extracts

When an extract refresh is performed on extracts created in Tableau 10.4 and earlier (that is, a .tde extract), the extract is upgraded to .hyper extract automatically. While there are many
benefits of upgrading to a .hyper extract, you will be unable to open the extract with previous versions of Tableau Desktop. For more information, see Extract Upgrade to .hyper Format.

Setting up refresh schedules

As a server administrator, you can enable scheduling for extract refresh tasks, and then create, change, and reassign schedules. General scheduling options you change on the server are available as part of the publishing process when a Tableau Desktop user publishes an extract.

Schedules that you create have the following options:

Priority

The priority determines the order in which refresh tasks are run, where 0 is the highest priority and 100 is the lowest priority. The priority is set to 50 by default.

Execution mode

The execution mode indicates to the Tableau Server backgrounder processes whether to run refreshes in parallel or serially. Schedules that run in parallel use all available backgrounder processes and serial schedules run on only one backgrounder process. However, a schedule can contain one or more refresh tasks, and each task will only use one backgrounder process, whether in parallel or serial mode. This means that a schedule in parallel execution mode will use all available backgrounder processes to run the tasks under it in parallel, but each task will only use one backgrounder process. A serial schedule uses only one backgrounder process to run one task at a time.

By default, the execution mode is set to parallel, so that refresh tasks finish as quickly as possible. You might want to set the execution mode to serial (and set a lower priority) if you have a very large schedule that prevents other schedules from running.

Frequency

You can set the frequency to hourly, daily, weekly, or monthly.

For information, see Create or Modify a Schedule.
Refreshing extracts manually

In the Tableau Server web environment, both server and site administrators can run extract refreshes on-demand on the Schedules page:

- Select the schedule and click Actions > Run Now.

You can also refresh extracts from the command line using the tabcmd refreshextracts command. For more information, see tabcmd Commands.

Refreshing extracts from Tableau Desktop

Tableau Desktop users can refresh extracts they publish and own. They can do this the following ways:

- **At publish time**: When an author publishes a workbook or data source that uses an extract, that author can add it to server refresh schedule. The refresh can be a full or an incremental refresh.

  Incremental refreshes reference a column in the extract that has a data type of date, date/time, or integer; such as a timestamp. Tableau uses this column to identify new rows that need to be added to the extract. For more information, see Refreshing Extracts and Schedule Extract Refreshes as You Publish a Workbook in the Tableau Help.

- **User interface**: In Tableau Desktop, you can use the Refresh from Source, Add Data From File, and Add Data From Data Source commands to upload an addition to or refresh an extract on Tableau Server. A user might want to do this if Tableau Server doesn’t have sufficient credentials to access the underlying data. For more information, see Updating Extracts on Tableau Server in the Tableau Help.

- **Data Extract command line utility**: The Data Extract command line utility installs with Tableau Desktop. You can use it to append to or refresh a published extract. For more information, see Tableau Data Extract Command Line Utility in the Tableau Help.
Enable Extract Refresh Scheduling and Failure Notification

Your publishers can schedule extract refreshes when two conditions are met:

- Tableau Server is configured to send email messages when extract refreshes fail. This is configured by a Tableau Server Manager (TSM) administrator and is on by default. For details, see Configure Server Event Notification.
- The site or sites in which you want to allow publishers to schedule extract refreshes is configured to send email when the refresh fails. This is configured by a server administrator in Tableau Server and is on by default. The instructions below explain how to do this if it is not enabled.

While you’re enabling scheduling, you can decide whether also to enable sending email to owners of data sources or workbooks that are refreshed when those extract refreshes do not complete successfully. You can read more about these emails below. When you enable refresh failure notification, the owners of the content that has scheduled refreshes can opt out individually by changing their account settings.

1. Sign in to Tableau Server as a server administrator.

2. Go to the General tab of the Settings page for the site you want to configure for subscriptions:
   - If you have a single site, at the top the browser window, click Settings and General.
   - If you have multiple sites, select the site you want to configure and click Settings and General.

3. On the General page, do the following:
   - Under Refresh Failure Notifications, select Send email to data source and workbook owners when scheduled refreshes fail.

   To clarify, if a scheduled refresh for a particular data source fails, the email goes only to the owner of that data source, not to owners of workbooks that connect to that data source.
Under **Embedded Credentials**, select both options to let publishers embed credentials and schedule extract refreshes. (Automatic refresh schedules require embedded credentials so Tableau Server can directly access data.)

**Note:** On a multi-site server, failure notifications are a site setting, and embedded credentials are a server setting. To configure embedded credentials on a multi-site server, select **Manage All Sites** and then **Settings**.

**Managing schedules from the server**

In your organization it might be more appropriate to manage embedded credentials and refresh schedules centrally from the server. If you do that, you might clear the check boxes in the **Embedded Credentials** section described in the steps above, so that Tableau Desktop publishers do not see schedule options during publishing.

Managing schedules centrally enables you to distribute extract refresh and subscription tasks, so you can run them when most people are offline. It also enables you to oversee which credentials are embedded in connections.

For more guidelines for managing schedules and refreshes on the server, see **Provide access to data sources** and **Keep data fresh** in *Everybody's Install Guide*.

**How refresh failure emails work**

The email notification for a failed extract refresh lists the extract name and location on the server, gives the time of last successful refresh, the number of consecutive times the refresh has failed, and suggests the reason for the failure and possible solution.

After five consecutive failures, the refresh schedule is suspended until you or the data owner takes an action to address the cause of the failure, such as updating database credentials or a path to the original data file.
How the last successful refresh date is determined

The last successful refresh date and time are shown when that last refresh occurred within a number of days. By default it is 14 days, and this value is set in wgserv-er.alerts.observed_days. If the number of days since the last successful refresh exceeds the number specified in this setting, the message in the email shows “not in the last N days.”

Create or Modify a Schedule

The Schedules page shows a list of schedules, including their name, type, what they’re for (scope), number of tasks, behavior (concurrent or serial processing), and when they are scheduled to run.

**Note:** If you enable custom schedules for subscriptions on one or more sites, users will control schedules for their subscriptions on those sites. To learn more, see Enable Custom Schedules for Subscriptions.

To create a new schedule

1. In a site, click **Schedules**.

2. Click **New Schedule**.
3. Specify a descriptive **Name** for the schedule. For example, End of week.

4. Select a **Task type** the schedule will handle—refreshing extracts, running flows, or delivering subscriptions.

5. Optionally you can define a **Default priority** from 1 to 100, where 1 is the highest priority. This is the priority that will be assigned to the tasks by default. If two tasks are pending in the queue, the one with the higher priority runs first.

6. **Execution**: choose whether a schedule will run in parallel or serially. Schedules that run in parallel run on all available backgrounder processes so that they can complete faster.

   **Note**: Schedules for the same workbook will always run serially, even if you set this option to parallel.

7. Finish defining the schedule. You can define an hourly, daily, weekly, or monthly schedule.
8. Click Create.

To modify an existing schedule

1. Navigate to the Schedules page.

2. Select an existing schedule, click the Actions drop-down arrow, and then select Edit Settings.
3. Finish editing the schedule, and click **Save**.

**See also**

- Manage Refresh Tasks
- Extract Refresh Schedules
- [Schedule a flow task](#)

### Enable Custom Schedules for Subscriptions

Custom schedules for subscriptions allow users to receive email messages on a schedule that they define, rather than using a fixed schedule defined by a server administrator. Custom schedules for subscriptions have been available to users of Tableau Online since March 2017, and these schedules can now be enabled on a per-site basis in Tableau Server version 2018.2. Enabling custom schedules for subscriptions is a permanent change on any sites where you make this change. Any sites where you don’t enable custom schedules remain on fixed schedules defined by a server administrator.
Enable custom schedules

Before you can enable custom schedules on one or more sites, you must first enable custom schedules on Tableau Server, and then enable custom schedules on one or more sites on that server. To learn more about enabling subscriptions on Tableau Server, see Set Up a Site for Subscriptions.

**Step 1: Enable custom schedules on Tableau Server**

From a command prompt with Tableau administrator permissions, run the following commands:

```
tsm configuration set -k features.SelfServiceSchedules -v true
```
```
tsm pending-changes apply
```

This operation will restart Tableau Server.

**Step 2: Enable custom schedules on a site**

1. Log in to Tableau Server as a server administrator using a web browser:

   https://<hostname>/#/login

2. Browse to the Site Settings page for a site, and then enable custom schedules:
   1. Click **All Sites**, and then choose one of the sites from the drop-down list.
   2. Click **Settings**.
   3. On the **General** tab, under **Subscriptions**, select the following checkbox: **Permanently convert from fixed schedules created by administrators to custom schedules created by users. (You can’t undo this)**.
   4. Click **Save**.

When custom schedules are enabled on a site, all fixed schedules with a subscription are converted to an equivalent custom schedule.
How Scheduled Server Tasks are Prioritized

When processing scheduled extract refreshes and subscriptions, Tableau Server prioritizes background tasks in this order:

1. Any task already in process is completed first.

2. Any task that you initiate manually using Run now starts when the next backgrounder process becomes available.

3. Tasks with the highest priority (the lowest number) start next, independent of how long they have been in the queue.

   For example, a task with a priority of 20 will run before a task with a priority of 50, even if the second task has been waiting longer. To change task priority, see Create or Modify a Schedule.

4. Tasks with the same priority are executed in the order they were added to the queue. The first task added to the queue starts first; then the second task starts.

5. When multiple tasks with the same priority are scheduled to run at the same time, they start in the order they were created or enabled. Tasks scheduled for the same time are executed by job type with the fastest category of jobs starting first: subscriptions, then incremental extracts, then full extracts.

The following limitations also impact scheduled tasks:

- The number of concurrent tasks is limited to the number of backgrounder processes you have configured for Tableau Server.

- Separate refreshes for the same extract cannot run at the same time.

- Tasks associated with a schedule that is set to run serially run one at a time.
Configure Workbook Performance after a Scheduled Refresh

To improve the load times for workbooks, Tableau Server caches the results of queries included in workbooks. For most workbooks, query results are computed and cached when they are first viewed by a user on Tableau Server. However, for workbooks that connect to data extracts, Tableau Server can recompute query results when the corresponding extract refresh tasks run. This reduces the load time for these workbooks when they are first viewed, so this option is turned on by default for workbooks that have been viewed recently.

Determine the performance impact

Although this option reduces the initial load time for workbooks, recomputing query results also increases the load on Tableau Server. If your Tableau Server installation is already performance-constrained, you might want to turn this option off or lower the threshold for workbook caching.

Here are some possible reasons why you might want to turn this option off or lower the threshold:

- The Background Tasks for Non Extracts administrative view displays many long-running jobs in the Warming up external query cache on data change category.
- The Background Task Delay administrative view displays long delays.
- CPU and memory consumption for the backgronder processes is consistently high.

However, note that this is only one of the options that impacts the performance of background tasks. For more information about performance, see Performance.

Turn off workbook caching for the server

To decrease the load on Tableau Server, you can turn off workbook caching after a scheduled refresh at the server-level. If you turn this option off, Tableau Server caches query results for workbooks the first time the workbooks are viewed.
Use the following tsm configuration set option to turn off workbook caching after a scheduled refresh:

```
backgrounder.externalquerycachewarmup.enabled
```

For more information on how to use and apply tsm set options, see tsm configuration set Options.

**Turn off workbook caching for a site**

You can also turn off workbook caching after a scheduled refresh for an individual site. For example, you might do this if there is one site in particular that contains many slow workbooks which increase load on the server.

1. Select the site for which you want to turn off workbook caching in the sites drop-down.
2. Click Settings.
3. In the **Workbook Performance after a Scheduled Refresh** section, clear the check box.

**Note:** Although this option is available in the settings for an individual site, you must have server administrator permissions to view it.

**Configure the workbook caching threshold**

Tableau Server only recomputes query results for workbooks that both have scheduled refresh tasks and have been viewed recently.

You can increase or decrease the number of workbooks that are cached after a scheduled refresh with the following tsm configuration set option:

```
backgrounder.externalquerycachewarmup.view_threshold
```

By default, the threshold is set to 2.0. The threshold is equal to the number of views that a workbook has received in the past seven days divided by the number of refreshes scheduled
in the next seven days. (If a workbook has not been viewed in the past seven days, it is unlikely that it will be viewed soon, so Tableau Server does not spend resources recomputing queries for the workbook.)

Ensure Access to Subscriptions and Data-Driven Alerts

To ensure that users see the Subscribe and Alert buttons in the Tableau Server toolbar and can receive related emails, do the following:

- **Configure SMTP and event notifications on Tableau Server**: See Set Up a Site for Subscriptions.

- **Ensure that users have an email address in Tableau Server**: Users can update their email address on their account settings page.

- **Embed database credentials or don’t require them**: To email data in a view, Tableau Server needs to access the data without user involvement. This can be accomplished by using a workbook with embedded database credentials, a Tableau Server data source, or data that doesn’t require credentials (such as a file that’s included with the workbook at publish time).

- **Ensure that users can access needed workbooks and views**: Access to content on the server is controlled by the View permission. To receive images of content in email messages, users also need the Download Image/PDF permission. For more information, see Permission capabilities.

- **Avoid trusted authentication for embedded views**: If you use restricted tickets (the default) to render an embedded view, the Subscribe and Alert buttons don’t appear.

Set Up a Site for Subscriptions

When users subscribe to a workbook or view, a snapshot of the view is email to them on scheduled basis, so they can see the latest updates without having to sign into Tableau Server. Administrators, project leaders with appropriate site roles, and content owners
have the option to subscribe other users to workbooks and views. For more information, see Subscribe to Views.

**Note:** To create and receive subscriptions, users need access to related databases and views. See this list of requirements for details.


Prerequisite: Configure the server to send subscription emails

Before you can enable subscriptions for a site, you need to complete the steps to enable subscriptions on the server. Follow the steps in these topics to configure subscriptions on the server.

1. Configure SMTP Setup
2. Configure Server Event Notification

Enable subscriptions

After you have configured SMTP and server event notifications, you can enable subscriptions.

To enable subscriptions:

1. Sign into Tableau Server as a server administrator.
2. Go to the General tab of the Settings page for the site you want to configure for subscriptions:
   - If you have a single site, on the side navigation, click Settings and General.
   - If you have multiple sites, select the site you want to configure and click Settings and General.
3. On the General tab, scroll down to Subscriptions and select the subscription options
for your users.

**Note:** This section is only visible when the server-wide configuration option *Allow users to receive email for views that they have subscribed to* is enabled. For details, see Configure Server Event Notification.

a. Select *Allow users to subscribe to workbooks and views*

b. (Optional) To allow content owners to subscribe other users to their content, select *Allow content owners to subscribe other users to workbooks and sheets*.

4. (Optional) Scroll down to **Email Settings**.

   a. Enter an **Email From Address** that will show as the "From" address in email messages.

   b. Enter an **Email Footer** for email messages.

   A site's "From" address and message footer are also used in emails for data-driven alerts.

5. Click **Save**.

To specify the subscription schedules available to users, see *Create or Modify a Schedule*.

**Test subscriptions in a site**

1. **Subscribe to a view.**

2. At the top the browser window, click **Schedules**.

3. Select the schedule you chose for the subscription, and then click **Actions > Run Now**.
A snapshot of the view should be emailed to you within 10 minutes. If you experience an issue, see Troubleshoot Subscriptions.

Manage all user subscriptions

1. At the top the browser window, click Tasks, and then click Subscriptions.

   All user subscriptions for the current site appear, including information like subscriber name, view name, and delivery schedule.

2. Select any subscription you want to update. From the Actions menu, select Change Schedule, Change Subject, Change Empty View Mode, or Unsubscribe.

   (The empty-view option sends subscription emails only when data exists in a view. It's a good choice for high-priority alerts.)

Suspended Subscriptions

By default, a subscription is suspended after 5 consecutive subscription failures and result in the subscription emails not sent. To change the threshold number of subscription failures that can occur before they are suspended, use the tsm configuration set option, backgrounder.subscription_failure_threshold_for_run_prevention. This sets the threshold for the
number of consecutive failed subscriptions necessary before suspending the subscription. This is a server-wide setting.

Only Server administrators can configure the threshold number of subscription failures before a subscription is suspended.

Server administrators can opt in to receive email notifications when a subscription is sus-
pended. You can do this by navigating to My account settings -> Subscription Noti-
fications. This setting is at a site-level so has to be configured for site separately.

Resume suspended subscriptions

If a subscription fails more than five times, you’ll receive a notification email that your sub-
scription has been suspended. There are a few ways to resume a suspended subscription if you’re a subscription owner or administrator:

- From the My Content area of Tableau web pages, an icon appears in the Last update column to indicate that the subscription is suspended. Select ... > Resume Subscription to resume.

- From the Subscriptions tab of the affected workbook, an icon appears in the last update column to indicate that the subscription is suspended. Select ... > Resume Subscription to resume.

- From the Subscriptions tab under Tasks, an icon appears in the last update column to indicate that the subscription is suspended. Select ... > Resume Subscription to resume (Server administrators only).

When a subscription is resumed, the alert failing count goes back to zero. The next eval-
uation of the subscription will occur at the next scheduled evaluation time.

See also

Subscribe to Views in the Tableau Desktop and web authoring Help.

Project-level administration to learn which site roles allow full Project Leader capabilities.
Set Up for Data-Driven Alerts

When data reaches important thresholds for your business, data-driven alerts automatically send email notifications to key people users specify. As a Tableau Server administrator, you set up data-driven alerts much like you do subscriptions. For information about how users create and manage these alerts, see Send Data-Driven Alerts in Tableau User Help.

**Note:** To create and receive data-driven alerts, users need access to related databases and views. See this list of requirements for details. If alerts are enabled for a site, any user on that site can create them.

**Configure email for data-driven alerts**

1. Complete the steps in Configure SMTP Setup so the server can send email.

2. While viewing a site, click **Settings** at the top of the browser window.

3. Under Email Settings, enter a site-specific "From" address or message footer.

   A site's "From" address and message footer are also used in emails for subscriptions.

4. Click **Save**.

**Manage all data-driven alerts in a site**

1. At the top the browser window, click **Tasks**, and then click **Alerts**.

2. Select any alerts you want to update.

3. From the **Actions** menu, do any of the following:

   - Add or remove yourself as a recipient.
   - Edit alerts to change data thresholds, delivery schedules, and the full list of
recipients.

- Change alert ownership to different users, or delete alerts.

Disable data-driven alerts for a site

Data-driven alerts are supported for all sites by default, but administrators can disable them for specific ones.

1. While viewing a site, click **Settings** at the top of the browser window.

2. Under Data-Driven Alerts, uncheck **Let users create alerts and receive alert emails**.

3. Click **Save**.

Suspend data-driven alerts

By default, an alert is suspended after 350 consecutive alert failures. Server administrators can configure the threshold number of alert failures before an alert is suspended. To change the threshold number of data-driven alert failures that can occur before alerts are suspended, use the `tsm configuration set` option, `dataAlerts.SuspendFailureThreshold`.

This sets the threshold for the number of consecutive failed alerts necessary before suspending the alert. This is a server-wide setting. The threshold value applies to every configured data-driven alert on the server.

Resume suspended alerts

If an alert fails enough times, you'll receive a notification email that your alert has been suspended. There are a few ways that administrators or alert owners can resume a suspended alert:
From the Tasks > Alerts area of Tableau web pages, an icon appears in the Last checked column to indicate that the alert is suspended. Select ... > Resume Alert to resume the alert.

- Click Resume Alert in the notification email to resume the alert. A notification will either allow you to resume the alert, or indicate that the view has changed and the alert should be deleted.

- From the Alerts panel of the affected view or workbook. To resume the alert from a view or workbook, select Alert to open the Alerts panel. An icon appears next to the suspended alert. Select Actions > Resume Alert on the affected alert to resume.

Alert owners will receive an email notification when the alert is working again.

Control how often the server checks data-driven alerts

By default, Tableau Server checks every 60 minutes to confirm whether data conditions for alerts are true. If you notice performance impacts, you can customize this time interval with the tsm configuration set option, dataAlerts.checkIntervalInMinutes.

Regardless of the dataAlerts.checkIntervalInMinute setting, the server also checks alerts whenever extracts in the related workbook are refreshed. To check an alert more frequently than the setting specifies, change the extract-refresh schedule.

Track the server's alert-checking process

In the Background Tasks for Non Extracts view, you can track the server's alert-checking process by looking for these tasks:

- Find Data Alerts to Check

- Check If Data Alert Condition Is True

The "Find" task limits "Check" tasks to alerts that can currently send related emails. For example, if a user has chosen an email frequency of "Daily at most", after the alert condition becomes true, the server waits 24 hours before checking the alert again.
Each "Check" task uses one server backgrounder process, loading the related view to evaluate the alert condition. If all users see the same version of a view, it loads only once. But if users have applied filters to a view, or the data they see is limited by user-level security, the view loads once for each recipient.

Identify and fix failing alerts

As an administrator, you can proactively identify failing alerts that users may be unaware of. To check:

1. Select Status in your site menu
2. Select Background tasks for non-extracts
3. From the Task drop-down menu, select Check if Data Alert is True
4. In the far right, click on Error to see a list of failing alerts
5. Hover over the red failure icon to display a tooltip with alert details

To determine the alert owner, look for the alert ID number in the data_alerts table of the Tableau Server Repository. (In the alert management area of a site, you can also look for the alert name following the number, but be aware that multiple different alerts may use the same name.)

Note: Starting in Tableau Server 2019.3, alert owners will be automatically notified when an alert fails ten times. Administrators can customize when alert owners receive notifications. Administrators can configure Users will not be notified for alerts that failed prior to upgrading to Tableau Server 2018.1.
Failing alerts are often caused by content changes on Tableau Server. Encourage users to recreate alerts if changes like the following occur:

- A workbook, view, or data field is removed or renamed.
- Database credentials embedded in workbooks expire. (Alerts require workbooks to use either embedded credentials, or none at all.)
- A data source becomes inaccessible.

**Tip:** To automatically get emailed when alerts fail, follow the steps in Collect Data with the Tableau Server Repository, and connect to the "background_jobs" table. From that table, create a custom view that includes the "Check If Data Alert Condition Is True" job name and its finish code. Then set up a data-driven alert to email you whenever a finish code equals 1 (failure).
Managing Jobs in Tableau Server

In Tableau Server, users can schedule extract refreshes, subscriptions, or flows to run periodically. These scheduled items are referred to as Tasks. The Backgrounder process initiates unique instances of these tasks to run them at the scheduled time. The unique instances of the tasks that are initiated as a result are referred to as Jobs. Jobs are also created for runs that are initiated manually, by clicking the Run now option.

For example, an extract refresh task is created to run daily at 9 AM. This is an extract refresh task, and every day at 9 AM, a job will be created for the Backgrounder to run. In addition to user-generated jobs, the Backgrounder also does a number of System jobs on behalf of the user to support general Tableau work flows, such as thumbnail generation.

Running all these jobs can mean that Backgrounder uses a lot of resources at various times during the day. Using the Job Management feature, Server and Site administrators can get more details on these jobs that happen in their Server or Site, and take action on those jobs to better manage server resource usage. System jobs are only viewable by Server administrators, and by default are filtered out.

The Jobs page which contains the information about jobs can be accessed by navigating to the Existing Tasks menu of the left navigation menu.

Information about jobs can only be viewed by Server and Site administrators.

Overview

This topic describes how to view and understand the information displayed in the Jobs page.

At the top of the page there are high level statistics for the number of Failed, Completed, and Cancelled jobs within the past 24 hours. For Server administrators, this also includes System jobs. Applying filters do not change these values.
For each job generated, there is a Job ID, the status of that job, the priority, the type of task that the job was generated from, the current run time - if the job is in-progress, current queue time - if queued, as well as the average run time, and average queue time. Tableau records historical run and queue times to compute the average run time and average queue time.

The Job ID can be useful when viewing jobs on Admin views and can also be used to query the Workgroups Database. When you click on the Job ID, you will see more detailed information about the job, such as the Job LUID, the project name, the schedule, the content name, content owner, job creator, the last time the job ran successfully, and the site name. The site name is displayed if you navigate to the Jobs page using the Manage All Sites menu.

**Job Details for Job 617**

- **LUID:** 0e2aede-71c7-4a56-8c2c-b03c47cc5b5a
- **Project:** Project1
- **Schedule:** Every Sunday - 4:00PM
- **Content:** Rust interpreter speed study
- **Content Owner:** Sabraan
- **Job Creator:** Andrea
- **Last Successful Run:** Sep 9, 2019, 7:45 AM
- **Site:** Site1

[OK]
**Note:** Doing a **Refresh Now** from the **Data Sources** page will only show the LUID information in the **Job Details** dialog box.

**Important!** Jobs that existed 24 hours or newer before an upgrade to Tableau Server 2019.4 will not have data for **Average Queue Time, Average Run Time, Last Successful Run Time**, and **Job Creator** on the **Jobs** page.

**Task Types**

There are five types of tasks:

- **Extracts:** This includes extract creation, incremental extract refreshes, and full extract refreshes. For more information on extract refreshes, see Quick Start: Refresh Extracts on a Schedule.

- **Subscriptions:** For more information, see Set Up a Site for Subscriptions.

- **Flow:** For more information, see Schedule a Flow Task.

- **Encryption:** Includes the following:
  - Extract encryption and decryption
  - Flow encryption and decryption
  - Re-key extracts and flows

- **System:** This is all system Jobs that the Backgrounder handles behind the scenes to support Tableau Server.

**Filters**

You can filter to see only certain jobs. The available filters are by Job Status type, Task Type, and Time Range. For the Time Range filter, you can choose from past one to 24 hours, in four hour increments. The option to filter in System Jobs is available if you are a Server Administrator.
Canceling Jobs

Extract refreshes, subscriptions and flow run jobs can be canceled. You can only cancel one job at a time, and selecting multiple jobs at one time for cancellation is not supported.

When you cancel a job, an email with the time the job was canceled, the affected content, and the time the job ran before being canceled is sent to the recipients that you select in the Cancel Job dialog box. In addition you can add your customized notes to be included in the email.

If you do not select any recipients, the job will be canceled, but no email will be sent.

To cancel a job, click on the ellipses next to the Job ID and use the dialog to cancel the job:
Status

There are seven types of status that jobs can be in, and hovering over each status will display more relevant information.

- **Completed**: This job shows as **Completed successfully** and you can see the time when the job completed in the tooltip that is displayed when you hover over the status.

```
[ ] Completed
```

```
This job successfully completed on Jun 13, 2019, 10:23 AM. Learn more
```

- **In Progress**: This job shows as **In Progress**. A time for how long the job has been
running for is displayed in the tooltip when you hover over the status.

- **In Progress**: This job is in **In Progress**, but is running late. Tableau keeps track of the average run times for the same job, and if the current run time is longer than the average run time, then it is considered running late. Times for how much longer than average the job has been running and its average run time is provided in the tooltip that is displayed when you hover over the status.

- **Pending**: This job is currently **Pending**, waiting to be run when there is available Backgrounder capacity. A time for how long the job has been in the queue for is provided in the tooltip that is displayed when you hover over the status.

- **Pending**: This job is currently **Pending**, but is running late. Tableau keeps track of the average queue times for the same job, and if the current queue time is longer than
the average queue time then it is considered running late. Times for how much longer than average the job has been queued is provided in the tooltip that is displayed when you hover over the status.

**Cancelled:** This job was **Cancelled** by a Server or Site administrator. The time the job was cancelled and how long it ran for before cancellation is provided in the tooltip that is displayed when you hover over the status.

**Failed:** This job is showing as **Failed**. The time when the job failed, how long it ran for before it failed, and why the job has failed is provided in the tooltip that is displayed when you hover over the status.

**Suspended:** This job is showing as **Failed** with a pause icon. If the job fails 5 times
consecutively, then the job is suspended. Suspended tasks are still available but Back-grounder will not create jobs for these tasks until they are resumed by the user.

Failed

This job did not run since it was initiated from a task that is suspended. To run jobs initiated from this task, you must first resume the task. Learn more

Administrative Views

The Status page contains an embedded Tableau workbook with various administrative views. These views help you to monitor different types of server or site activity.

Note: You must install PostgreSQL drivers before you can see Administrative views. For more information, see Database Drivers.

Navigating to administrative views

To see administrative views, click Status. Site administrators can see administrative views for their site. Administrators of multiple sites can see views for the current site.

On a multi-site server, server administrators can see views for the entire server. Click the site menu, and then click Manage All Sites to access the server menus.
To see views for individual sites on a multi-site server, click the site menu, select the site name, and then click **Site Status**.
Pre-built Administrative Views

Administrative views are powerful monitoring tools that can help you optimize Tableau Server and better understand how your users are interacting with Tableau content. The administrative views listed to the right are included with Tableau Server. Click on the link for a view to learn more about how to interpret and act on the information the view provides.

To create your own administrative view, see Create Custom Administrative Views.
Performance of Views

**Note:** This view is only available to server administrators. For information about how to navigate to administrative views, see Administrative Views.

The Performance of Views administrative view displays how long it takes for views to load and how many sessions are running at a time on the server.

You can compare spikes in the number of sessions with spikes in slow load times to identify the times of day when high user traffic is slowing down the server. You can also look at the individual views by load time to understand which views take the longest to load.

Some views might take a long time to load regardless of when they are viewed. You can identify which workbooks need to be optimized with the Stats for Load Times administrative view. Some simple ways to optimize workbooks includes the following:
Tableau Server on Linux Administrator Guide

- Display less information in each view.
- Break up views.
- Reduce the number of filters.
- Use data extracts.

Performance of Flow Runs

Use this view to see the performance history for all the flows on a site. You can filter by Flow Name, the Output Step Name, Flow Owner, Run Type (Scheduled or Ad Hoc), and the time the flow runs were started. For information about other administrative views available for flows, see Monitor Flow Health and Performance in the Tableau Prep help.

Here are some questions you can answer using this view:

- **What flow tasks are currently scheduled?** – To do this, use the Start Time filter and select the time frame you want to look at. For example, to see flow tasks that are scheduled in the next 3 hours, select **Hours -> Next ->** and enter 3.

- **What is the duration of flow tasks?** - To answer this, click on a mark in the view and you should see details including the task duration.

**How many flows were run ad hoc, and how many were scheduled runs?** - To answer this, use the **Run Type** filter and select **Ad hoc or Scheduled**.

**Note:** This is not functional in this release and will not actually filter the data.
In addition to the questions described above, here are some examples of insights you might be able to gather:

- The flows that are running most frequently will have the most marks.

- To see how many flows are running at the same time currently, hover over a mark that shows “In Progress” or “Pending and select “Keep Only” to filter all flow runs that are currently running.

- To see how many flows are running at the same time during a specific time range, select a range for the Start Time filter. For example, you can choose “Next three hours” to see which flows will be running in the next three hours.

Traffic to Views

The Traffic to Views view gives you the ability to see how much of your user traffic goes to views.

You can filter what information is displayed and the time frame it comes from by selecting the view, the workbook, and the time range. Server administrators can specify the site.
Two time lines at the top of the view show you how views are being used over a time range you specify (the default is the last 7 days):

- **What is the Total View Count by Day**—This shows total view count by day, based on the filters you set. Hover your mouse pointer over a point on the line to see the count of views. Select the point to update the other sections of the view based on your selection.
- **What is the Total View Count by Time**—This shows the view count by time of day. The filters and any selection impact this graph.

Two bar graphs at the bottom of the view show results that are filtered by the **Min View Count** filter at the top of the view. These show you the views that are most often accessed, and the users who most frequently access views. Only those views and users with counts greater than or equal to the minimum view count value are displayed:

- **What Views are Seen the Most**—This is a list of the most visited views. Like the other sections of the view, the information is limited by filters and any selection you make.
- **Who Accesses Views Most Often**—This shows the users who most often access the views and is limited by filters and any selection you make.

**Traffic to Data Sources**

The Traffic to Data Sources view gives you the ability to see usage of data sources on your Tableau Server installation. This can help you determine which data sources are most heavily used and those that are less often used. You can filter the information you see by selecting the data source, the action taken on that data source, and the time range. Server administrators can specify the site.

A timeline at the top of the view shows you how data sources are being used over a time range you specify (the default is the last 7 days):

- **What is the Total Data Source Usage by Day**—This shows total data source usage by day, based on the filters you set. Hover your mouse pointer over a point on the line to see the count. Select the point to update the other sections of the view based on your selection.

Two bar graphs at the bottom of the view show results that are filtered by the **Min Interactions** filter at the top of the view. These show you which data sources are most used, and who uses data sources most often. Only those data sources and users with interaction counts greater than or equal to the minimum interactions value are displayed:
What Data Sources are Used Most—This is a list of the most used data sources. Like the other sections of the view, the information is limited by filters and any selection you make.

Who Uses Data Sources Most Often—This shows the users who most often use the data sources. This is impacted by filters and any selection you make.

Actions by All Users

The Actions by All Users view gives you insight into how your Tableau Server installation is being used. You can filter the view by actions and by time range. Server administrators can filter by site. The Total Users count shows the number of users who have performed an action. This value is not affected by any filtering. The Active user count shows the number of active users who have performed one of the selected actions.

Up to three separate groups of time lines show you how users are using Tableau Server over a time range you specify (the default is the last 7 days). If no actions are selected for a particular group, that group does not display. Possible groups are:
- **Access & Interactions**—This shows you sign in (log on) activity, view access and data source use.
- **Publish & Download**—This shows publishing and downloading of flows, workbooks and data sources.
- **Subscriptions**—This shows counts of subscription email sent for workbooks and views. It also shows the counts of flow runs.

Use the legend at the bottom to view a subset of the displayed actions. Click a single action to highlight the line for the action, or **Ctrl + Click** on multiple actions to highlight more than one. To clear the selection and display all the selected actions, click on any action in the legend.

**Actions by Specific User**

The Actions by Specific User view gives you insight into how individual users are working in your Tableau Server installation. You can filter the view by user name, actions, and time range. Server administrators on multi-site installations can filter by site.
Tableau Server on Linux Administrator Guide

Up to three separate groups of time lines show you how a selected user is using Tableau Server over a time range you specify (the default is the last 7 days). If no actions are selected for a particular group, or if no actions were taken, that group does not display. Possible groups are:

- **Access & Interactions**—This shows you sign in (log on) activity, view access and data source use. This means any interaction with a Data Source by the selected user, including extract refreshes scheduled by that user, or the user accessing a workbook that is associated with that data source.
- **Publish & Download**—This shows publishing and downloading of flows, workbooks and data sources.
- **Subscriptions**—This shows counts of subscription email sent for workbooks and views. It also shows the counts of flow runs.

A bar graph at the bottom of the view shows which items the selected user is using.

Use the legend at the bottom to view a subset of the displayed actions. Click a single action to highlight the line for the action, or **Ctrl + Click** on multiple actions to highlight more than one. To clear the selection and display all the selected actions, click on any action in the legend.

**Actions by Recent Users**

The Actions by Recent Users view shows you which signed-in users have been active on Tableau Server over the past 24 hours. This can be useful if you need to perform some maintenance activity and want to know how many and which users this will affect, and what they are doing on Tableau Server.

The view **Active, Recently Active**, and **Idle** users that are currently signed in to Tableau Server. For this view, an active user is one who took an action in the last 5 minutes, a recently active user is one who last took an action within 30 minutes, and an idle user is one who last took an action more than 30 minutes ago. The actions are displayed in the lower section of the view.
Select a user to see only the actions that user performed recently. Hover over an action to see details of the action.

Background Tasks for Extracts

The Background Tasks for Extracts view displays extract-specific tasks that run on the server.
Understand this view

To better understand this pre-built administrative view, make note of the following:

- The table, "What Extracts Ran on this Server," lists the extracts that ran in the time period specified in Timeline.
- You can click Success or Error to filter the table based on status.
- You can also click a specific task to update the "How Much Time did Extracts Take" graph for the selected task.
- The table, "How Many Extracts Succeeded or Failed," updates for the status (success or failure) of the task, but the count of extracts that succeeded or failed does not change.

Status

Tasks can have a status of success or error.

<table>
<thead>
<tr>
<th>Icon</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>✖️</td>
<td>Error—Server was unable to complete the task.</td>
</tr>
</tbody>
</table>
Tableau Server on Linux Administrator Guide

<table>
<thead>
<tr>
<th>Icon</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image.png" alt="Success" /></td>
<td>Success—Server completed the task.</td>
</tr>
</tbody>
</table>

See details about a task

For details on about the task, use your mouse to hover over the success or error icon.

<table>
<thead>
<tr>
<th>Task ID: 87134</th>
<th>StopLight</th>
</tr>
</thead>
<tbody>
<tr>
<td>Task: Refresh Extracts</td>
<td>StopLight</td>
</tr>
<tr>
<td>Status of Task: Success</td>
<td>StopLight</td>
</tr>
<tr>
<td>Created at: 5/30/2018 9:00:16 PM</td>
<td>StopLight</td>
</tr>
<tr>
<td>Started At: 5/30/2018 9:00:25 PM</td>
<td>StopLight</td>
</tr>
<tr>
<td>Completed At: 5/30/2018 9:00:51 PM</td>
<td>StopLight</td>
</tr>
<tr>
<td>Runtime: 26 sec</td>
<td>StopLight</td>
</tr>
<tr>
<td>Priority: 50</td>
<td>StopLight</td>
</tr>
<tr>
<td>Backgrounder: localhost</td>
<td>StopLight</td>
</tr>
<tr>
<td>Backgrounder ID: localhost:101</td>
<td>StopLight</td>
</tr>
</tbody>
</table>

Finished refresh of extracts (new extract id [B5E45D13-C08B-454E-8023-93142472C981]) for Workbook StopLight

Background Tasks for Non Extracts

The Background Tasks for Non Extracts view displays tasks that the server runs that are not related to refreshing extracts. For example, edited OAuth connections, subscription notifications, and so on.
A table lists the tasks that ran in the time range specified. Click **Success** or **Error** to filter the table based on status. Select a specific task in the **How Many Tasks Succeeded or Failed on this Site** table to update the **What Background Tasks Ran on this Site** graph for the selected task.

Tasks can have a status of success or error. For details about the task, use your mouse to hover over the success or error icon.

<table>
<thead>
<tr>
<th>Icon</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>✖️</td>
<td><strong>Error</strong>—Server was unable to complete the task.</td>
</tr>
<tr>
<td>🔄</td>
<td><strong>Success</strong>—Server completed the task.</td>
</tr>
</tbody>
</table>

**Upgrade Thumbnails Job**

In Tableau Server version 2019.1 and later, workbooks and views use higher resolution thumbnails. When you upgrade to Tableau Server version 2019.1 or later, the Backgrounder
runs the Enqueue Thumbnails Upgrade job and the Upgrade Thumbnails job, which refreshes thumbnails for any views that have not been edited or published since Tableau version 2018.2 and converts them from low resolution thumbnail images (192 x 192 pixels) to high resolution thumbnail images (300 x 300 pixels), as shown in the figure below.

When the Upgrade Thumbnails job refreshes the thumbnail images, the modified date of the view is changed to match the thumbnail update date. The Upgrade Thumbnails job does not change the modified date of workbooks when it updates the contained view's thumbnails.

Depending on the number of workbooks you have, the Upgrade Thumbnails job can run for a few minutes to a few hours. You may notice that CPU usage on Backgrournder nodes is higher than normal during the time that the Upgrade Thumbnails job is running. You may also notice that thumbnail images may appear in low resolution for up to a week after upgrading to Tableau Server 2019.1 or later, while the Upgrade Thumbnails job processes all of the thumbnail images used in your workbooks. We recommend that you wait for the Upgrade Thumbnails job to complete before you back up Tableau Server.

After the first run of the Upgrade Thumbnails job, it runs on a predefined weekly schedule. The Upgrade Thumbnails job runs at lowest priority and creates one task per workbook in the background_jobs table to upgrade any low resolution thumbnails. Low resolution thumbnails published to Tableau Server by Tableau Desktop version 2018.3 and earlier are automatically cleaned up each week when the Upgrade Thumbnails job runs.
Troubleshooting

You can check the status of the Upgrade Thumbnails job using the Background Tasks for Non Extracts administrative view. The Upgrade Thumbnails job can display either the success or error status.

**Upgrade Thumbnails job failed, or it completed but some thumbnails are still low-resolution.**

The Upgrade Thumbnails job might show error status if your credentials are wrong. In that case, the workbook thumbnails will still appear in fuzzy, low-resolution (192 x192 pixels). Update your credentials, and the Upgrade Thumbnails job will update the workbook thumbnails the next time it runs.

**Background Task Delay**

**Note:** This view is only available to server administrators. For information about how to navigate to administrative views, see Administrative Views.

The Background Task Delay view displays the delay for flow tasks, extract refresh tasks and for subscription tasks—that is, the amount of time between when they are scheduled to run and when they actually run. You can use the view to help you identify places you can improve server performance by distributing your task schedules and by optimizing tasks.
Here are possible reasons for the delays, and ways that you might reduce the delays:

- Many tasks are scheduled for the same time. In the example view, tasks that show long delays are clustered at the same time every day, which creates spikes in the wait time. Note that you can set the Timeline filter to a single day to view task delays by hour and identify the hours of the day which have many tasks scheduled at the same time. A solution to this issue can be to distribute the tasks to off-peak hours to reduce load on the server.

- Specific tasks take a long time to run and are preventing other tasks from running. For example, there might be an extract refresh job that is connecting to a slow data source or that is processing a large amount of data. Use the Background Tasks for Extracts administrative view to identify which extract refresh tasks are running slowly. You can then optimize the extract refresh task by filtering the data, aggregating the data, or creating multiple data sources for individual tables in a data source.
Other server processes are running at the same time and are consuming server resources and slowing down performance. Monitor the CPU and memory usage of server processes to see which processes are consuming the most resources and then adjust the configuration of processes on your server.

Stats for Load Times

The Stats for Load Times view shows you which views are the most expensive in terms of server performance. You can filter by view and time range. Server administrators can filter by site. You can also limit the view based on load time in seconds, using the sliding Load Time filter. Load times are for the server. Depending on your client browser and networking, actual load time may vary slightly.

The **Average Load Times** graph shows average load times for views based on the filters you set. Hover over a point to see details. Select a point on the line to update the rest of the view for the selection:
The **Exact Load Times** view shows exact time to load the listed views. A vertical line shows the average load time for each view. Select a mark to see details of a specific instance of the view loading:

```
What are the Exact Load Times for Views?

<table>
<thead>
<tr>
<th>View Name</th>
<th>Site</th>
<th>Start Time</th>
</tr>
</thead>
<tbody>
<tr>
<td>Census2013MarchAge</td>
<td>Default</td>
<td>2014-03-15</td>
</tr>
<tr>
<td>DemoWorkbookPercentage</td>
<td>Default</td>
<td>2014-03-15</td>
</tr>
<tr>
<td>SoccerPlayerTransfersStory</td>
<td>Default</td>
<td>2014-03-15</td>
</tr>
<tr>
<td>MovieProductionHistoryMovie</td>
<td>Default</td>
<td>2014-03-15</td>
</tr>
<tr>
<td>SanFranciscoRealEstateMap</td>
<td>Default</td>
<td>2014-03-15</td>
</tr>
<tr>
<td>RegionalSelectUsersDemoDial</td>
<td>Default</td>
<td>2014-03-15</td>
</tr>
<tr>
<td>BusinessOverviewBusinessC</td>
<td>Default</td>
<td>2014-03-15</td>
</tr>
<tr>
<td>EbolaCountryDataMap</td>
<td>Default</td>
<td>2014-03-15</td>
</tr>
<tr>
<td>MarketHealthViewMap</td>
<td>Default</td>
<td>2014-03-15</td>
</tr>
<tr>
<td>SeattleWeatherRain</td>
<td>Default</td>
<td>2014-03-15</td>
</tr>
<tr>
<td>TaxiAnalysis/First</td>
<td>Default</td>
<td>2014-03-15</td>
</tr>
</tbody>
</table>

Stats for Space Usage

The Stats for Space Usage view can help you identify which workbooks and data sources are taking up the most disk space on the server. Disk space usage is displayed by user, project, and by the size of the flow output, workbook or data source and is rounded down to the nearest number:
Use the **Min Size** filter to control which data sources and workbooks are displayed, based on the amount of space they take up.

Three bar graphs give you information about space usage on your Tableau Server:

- **What Users Use the Most Space**—This shows the users who own data sources and workbooks that are taking up the most space. Click a user name to filter the next two graphs for that user. Click the data source bar or the workbook bar for a user to filter the next two graphs for that type of object for that user. Click the selected user or bar to clear the selection.

- **What Projects Use the Most Space**—This shows the projects with the data sources and workbooks that are using the most space. If a user or object type is selected in the What Users Use the Most Space graph, this displays information specific to the selection.

- **What Workbooks, Flows, and Data Sources Use the Most Space**—This shows
the workbooks and data sources that are taking the most space. The bars are color-coded based on the length of time since the last refresh.

Move your cursor over any bar to display usage details:

Click on a bar to select it and update the other areas of the view based on that selection.

Server Disk Space

**Note:** This view is only available to server administrators. For information about how to navigate to administrative views, see Administrative Views.

Use the Server Disk Space view to see how much disk space is in use on the computer or computers that run Tableau Server, where disk space refers only to the partition where Tableau Server is installed. You can also use this view to identify sudden changes in disk space usage.

For a distributed installation, the view displays information about each computer in the cluster.
The Server Disk Space view includes two graphs:

- **What is the most recent disk space usage?**—This graph shows disk space usage for the last 30 days both in gigabytes and as a percentage. Disk space refers only to the partition where Tableau Server is installed.
- **How has free disk space trended in the last month?**—This graph shows changes to disk space usage over the last month. Rest your pointer on a line to view the exact amount of free disk space for a point in time.

When Tableau Server is low on disk space, you can remove files to free space.

**Tip:** You can have Tableau Server notification you when free disk space falls below a threshold that you specify. For more information, see Configure Server Event Notification.
Desktop License Usage

Note: This view is only available to server administrators. For information about how to navigate to administrative views, see Administrative Views.

The Desktop License Usage view lets server administrators see usage data for Tableau Desktop licenses in your organization. This can help you manage licenses efficiently and determine if you need more or fewer licenses. This view can help you answer the following questions:

- Who is using a Tableau Desktop license in my enterprise?
- Have any licenses been shared or transferred?
- Is any license being used on a computer where it should not be?
- Does a specific user use their license?
- What types of licenses are being used in my enterprise?
- Do I need to convert any trial licenses?

Note: To get data about licenses, each copy of Tableau Desktop version 10.0 or later needs to be configured to send data to Tableau Server. This configuration can be done at installation time, using scripting or third-party software to install and configure Tableau, or after installation, by modifying the registry or property list file. For more information, see Configure Desktop License Reporting.

In order to view license data, Desktop License Reporting must be enabled on Tableau Server. See Enable and configure Desktop license reporting.

Filters

- Product Keys. Type a string to filter the dashboard to only those licenses that
include the string anywhere in the product key. For example, to see only licenses that begin with TDTD, type TDTD and press Return to filter the view. Click the X after the string to reset the filter.

- **Action.** Use this filter to control what the dashboard displays, based on the action taken. Actions are **Activate, Use**, and **Return** (deactivate). If the **Use** action is not selected, nothing is displayed in the top bar graph.

- **Department.** Use this filter to control what departments the dashboard displays licenses for. The filter is populated based on the **Department** values specified when Tableau Desktop is registered.

- **Select time duration in days.** Use this slider to specify the time length in days that the dashboard displays information for. The default value is 183 days.

When you hover over the filter card in the first three filters, a drop-down icon appears. Click the icon to specify whether the view should include data that matches the filter (the default) or exclude data that matches the filter:

![Include values Exclude values](image)

**Who has used Tableau in the last <nn> days?**

This area of the dashboard shows a bar graph of three types of Tableau Desktop licenses (Perpetual, Trial, and Term) and the number of users who have used each license type during the specified time period. Hover over a license type segment to see an explanation of the license type. Click a segment to filter the rest of the dashboard for only that license type. This action filters both the tables that show licenses that have been used and those that have not been. For example, to see a list of term licenses that have been used during the time period, click the Term bar. The "used" and "not been used" lists are filtered to just show term licenses.
A table of detailed information shows under the bar graph. For each row in the table, action icons display on the right, above a timeline that shows you when the action last took place.

To see a list of the underlying data in a format that allows you to select and copy values like email or product key, click a row in the list of licenses and click the View Data icon:

The data displays in summary form. Click Full data to see all the data. From this view you can select and copy individual values, or download the data as a text file.

What licenses have not been used in the last <nn> days

This area of the dashboard shows a list of licenses that have not been used during the specified time period. A timeline shows the last use date. Hovering over a last use mark gives you information including the registered user of the copy of Tableau.

Desktop License Expiration

Note: This view is only available to server administrators. For information about how to navigate to administrative views, see Administrative Views.

The Desktop License Expiration view gives server administrators information about which Tableau Desktop licenses in your organization have expired or need maintenance renewal. This can help you manage licenses efficiently. This view can help you answer the following questions:

- What trial or term licenses have expired?
- What perpetual licenses have expired maintenance?
- What perpetual licenses have maintenance renewals coming up?
To renew a license or get additional renewal information, see How to renew your Tableau licenses.

**Note:** In order to get data about licenses, each copy of Tableau Desktop version 10.0 or later needs to be configured to send data to Tableau Server. This configuration can be done at installation time, using scripting or third-party software to install and configure Tableau. For more information, see Configure Desktop License Reporting.

In order to view license data, Desktop License Reporting must be enabled on Tableau Server. See Enable and configure Desktop license reporting.

**Filters:**

- **Product Keys**—Type a string to filter the dashboard to only those licenses that include the string. For example, to only see licenses that begin with TDTD, type TDTD and press return to filter the view. Click the "x" after the string to reset the filter.

- **Department**—Use this filter to control what department(s) the dashboard displays licenses for. The filter is populated based on the Department values used when registering copies of Tableau Desktop.

- **Time Duration**—Use this filter to control the length of time for which the dashboard displays information.

The view includes the following tables, which are affected by the filters you set at the top of the view:

- **What keys have expired maintenance**—This table shows the product keys for which maintenance has expired, with a vertical line indicating the point at which the six month window for renewing maintenance closes. If maintenance for a key is expired for more than six months you need to purchase a new key in order to qualify for
support or upgrades.

- **What trial and term licenses have expired**—This shows the trial or term product keys that have expired.
- **What is the maintenance schedule for my keys**—This shows the keys and their maintenance status.

Create Custom Administrative Views

In addition to the pre-built administrative views available on the Maintenance page on the Server, you can use Tableau Desktop to query and build your own analyses of server activity. To do this, you can connect to and query views in the Tableau Server repository using one of two built-in users: the "tableau" or "readonly" user.

To connect to the Tableau Server repository, see Collect Data with the Tableau Server Repository.

- The **tableau** user—The tableau user has access to special views and a subset of tables in repository database. These views and tables are provided so that administrators can create custom administrative views. Tableau makes an effort to limit changes to these tables and views so that custom views built with them do not break.

- The **readonly** user—The readonly user has access to a large number of the repository tables, providing more data about server usage. Administrators can use these to create custom administrative views too, but many of the tables are intended primarily to support the functioning of Tableau Server and may be changed or removed without warning. This means that views created from these tables can break when the database structure is changed.

For examples of custom administrative views, see the Tableau Community. You can also use the temporary workbook that is generated when you view the build-in Administrative views.

Before you can connect using one of the built-in users, you must enable access to the Tableau Server database. After doing this you can use Tableau Desktop to connect to and query the database as the tableau user or the readonly user.
The `tsm configuration set option auditing.enabled` controls whether Tableau Server collects historical user activity and other information in the repository. It is enabled by default. Be aware that collecting historical events impacts the size of Tableau Server's backup file (.ts-bak).

- All `hist_tables` are controlled by the `tsm configuration set option wgserv-er.audit_history_expiration_days`, which controls how many days of event history are kept in the repository and has default value of 183 days.

- The `_http_requests` table is cleaned of all data older than 7 days when you run `tsm` maintenance cleanup. For more information, see Remove Unneeded Files.

- The `_background_tasks` table is cleaned automatically and keeps data for the last 30 days.

- All other tables with names that begin with a "_" prefix contain current data.

For more information about the tables in the Tableau Server repository, see Workgroup Database Data Dictionary.

**Performance**

You can monitor and tune the performance of Tableau Server.

**Tableau Server Performance Overview**

When you take the time to understand the performance of Tableau Server, you make it easier to serve your users by improving the efficiency of Tableau Server. Although every server environment is unique, and there are many variables that can impact performance, the general steps that you take to understand and act on performance data in Tableau Server are the same.
- **Notifications.** Configure email notifications for important server events. For example, you can receive notifications when server processes become unavailable and when the server is running out of disk space.

- **Monitoring.** Collect and analyze data about Tableau Server to understand how well the server is performing.

- **Tuning.** Make adjustments to tasks, process configurations, and more to improve the performance of Tableau Server.

- **Troubleshooting.** Identify bottlenecks in resources, workbooks, and more to improve the performance of Tableau Server.

### General Performance Guidelines

**Hardware and Software**

**Add more cores and memory:** Regardless of whether you’re running Tableau Server on one computer or several, the general rule is that more CPU cores and more RAM will give you better performance. Make sure you meet the Tableau Server recommended hardware and software requirements.

If you are running Tableau Server in a virtual environment, use your VM host’s best practices for vCPU allocation in relation to the number of physical CPU cores on the VM host.

**Configuration**

**Schedule refreshes for off-peak hours:** Backup tasks tend to stall other background tasks until the backup is completed. Use the Background Tasks for Extracts administrative view to see your refresh and backup task schedules. Your refresh tasks should be scheduled for off-peak hours that don’t overlap with your backup window.

**Look at caching:** Caching helps Tableau Server respond to client requests quickly, especially for views that connect to live databases. Use the tsm data-access caching list command to confirm the caching frequency is set to low (this is the default).
Consider changing two session memory settings:

- **VizQL session timeout limit**: The default VizQL session timeout limit is 30 minutes. Even if a VizQL session is idle, it is still consuming memory and CPU cycles. If you can make do with a lower limit, use `tsm configuration set Options` to change the `vizqlserver.session.expiry.timeout` setting.

- **VizQL clear session**: By default, VizQL sessions are kept in memory even when a user navigates away from a view. This reduces the need to rebuild views but consumes more session memory. To free up memory, you can end sessions when users leave views by changing the value of the `vizqlserver.clear_session_on_unload` setting to true. (Regardless of this setting, sessions for the Tableau Mobile app are always kept in memory, improving mobile performance.)

**Assess your process configuration**: Tableau Server is divided into six different components called server processes. While their default configuration is designed to work for a broad range of scenarios, you can also reconfigure them to achieve different performance goals. Specifically, you can control on which computers the processes run and how many are run. See Performance Tuning Examples for general guidelines for one-, two-, and three-node deployments.

**Server Resource Manager (SRM)**

The Server Resource Manager (SRM) monitors the system resources each Tableau process is using as well as tracking the total usage of Tableau Server on the system. If either a specific process or the product as a whole takes up too much system resources, SRM can notify the processes to free the resources or restart those processes.

The thresholds that determine when SRM will notify or restart a process are set in the SRM configuration options. The Tableau development team has set the default settings based on internal testing and don’t recommend you change these settings directly.

If you are seeing excessive system resource usage, we recommend that you contact Tableau Support to help determine if these configuration options need to be modified to solve the problem or issue that you are seeing.
Performance Monitoring Overview

When you monitor a server, you collect and analyze data that signals whether the server is performing badly or running into problems. For example, if you notice that your server is using 100% of its processing capacity for long periods of time, you know that there’s a problem.

The data that you need to collect and analyze can be broken down into the following broad categories:

- Resource usage data—how Tableau Server uses hardware resources like diskspace, memory, and processors.

- Session and load time data—how users interact with Tableau Server, including how long it takes for views to load and how many concurrent users there are.

- Background task data—how Tableau Server runs tasks that are not directly tied to a user action. For example, background tasks include extract refresh tasks, subscription tasks, and more.

Some of this data, including load time data and extract refresh data, is already accessible from the administrative views that are built into Tableau Server. However, to collect resource usage data you need to use a external performance monitoring tools. To collect additional load time data and background task data, you can connect to the Tableau Server repository.

For more information on the built-in administrative views, see Administrative Views.

**Note:** To use the sample workbook and to publish views to Tableau Server, you must have Tableau Desktop.
Collect Data with the Tableau Server Repository

The Tableau Server repository is a PostgreSQL database that stores data about all user interactions, extract refreshes, and more. You can enable access to the repository and use the data in it to help analyze and understand Tableau Server performance.

Looking for Tableau Server on Windows? See Collect Data with the Tableau Server Repository.

After you enable access to the Tableau Server repository, you can create views with data from the repository. The views that you create with this data are sometimes called custom administrative views. In addition to being used for performance monitoring, custom admin views can be used for tracking user activity, workbook activity, and more. For more information on the type of data that you can use for these views, see Create Custom Administrative Views and Workgroup Database Data Dictionary. Alternatively, if you are only interested in performance data, you can use the preselected database tables in the sample performance workbook.

Enable access to the Tableau Server repository

You can use Tableau Desktop to connect to and query the Tableau Server repository using two built-in users. The user named tableau has access to several database views you can use as part of building your own analyses of Tableau Server activity. The user named readonly has access to additional database tables that you can use to create views for even more in-depth analysis and this is the user we recommend you use.

Before you can connect to the repository, you need to enable access for the readonly user to the database. Use the tsm data-access repository-access enable command to enable repository access. When you enable repository access, you also create a password for the readonly user. You will use this password to access to the repository. You may also need to have port 8060 opened on the repository node so you can connect to the database.
1. Verify that port 8060 is opened on the computer where the repository is installed. This is a requirement if you are connecting remotely.

2. Enable repository access and create a readonly user password:

   ```
   tsm data-access repository-access enable --repository-user-name readonly --repository-password <PASSWORD>
   ```

   If your password includes special characters, you may need to escape the characters or enclose the password in quotes. Refer to the documentation for the Linux distro that you are running for information about passing special characters in bash shell.

   This command will restart Tableau Server.

   **Note:** If you later decide that you want to disable remote access to the Tableau Server repository, use the `tsm data-access repository-access disable` command. The command disables external access to the repository. This will not disable access from localhost. For more information, see `tsm data-access repository-access disable`.

---

**Connect to the Tableau Server repository**

This section describes how to connect to a custom set of tables from Tableau Server repository. For more information on the tables that you can connect to, see [Workgroups Database](#).

1. In Tableau Desktop select **Data > Connect to Data**, and then select **PostgreSQL** as the database to connect to.

   **Note:** You might need to install the PostgreSQL database drivers. You can download drivers from [www.tableau.com/support/drivers](http://www.tableau.com/support/drivers).
2. In the PostgreSQL connection dialog box, enter the name or URL for Tableau Server in the **Server** box. If you have a distributed server installation, enter the name or IP address of the node where the repository is hosted. Connect using the port you have set up for the `pgsql.port`, which is 8060 by default.

3. Specify **workgroup** as the database to connect to.

4. Connect using the user and the password you specified.

5. Click the **Require SSL** option if you have configured Tableau Server to use SSL for connecting to the repository. For more information, see [Configure Postgres SSL to Allow Direct Connections from Clients](#).

6. Click **Connect**.

7. Select one or more tables to connect to.
The tableau user has access to all of the tables that start with an underscore or with hist_. For example, you can connect to _background_tasks and _data-sources. The hist_ tables include information about server users that isn’t currently presented in the Actions by Specific User view. The readonly user has access to additional tables that can be used to query other information about server usage.

8. Click **Go to Worksheet**.

About the Tableau Server Data Dictionary

The Tableau Server Data Dictionary includes information about the tables and views in the "workgroup" PostgreSQL database of the Tableau Server repository. This database provides persistent storage for Tableau Server and is primarily intended to support that application. The Data Dictionary is not a complete description of all tables and fields in the database, and is provided for customers who want to query the database for information about usage on Tableau Server. Because the database and its contents are intended to support Tableau Server, the structure and contents may change without warning. This means any custom views you build from directly querying the database could break.

Open the Data Dictionary (new window).

Performance Tuning

This section describes how to use the performance data that you collect to identify ways to improve the performance of Tableau Server. Because no two server environments are identical, we can’t provide hard and fast rules for tuning server performance. However, you can draw conclusions about performance from patterns in the data that you collected.
For example, are there recurring spikes? Do any of the patterns that you notice in the administrative views correspond to similar patterns in a monitoring tool? Observing patterns like this can guide you in testing and incremental tuning.

Most performance tuning for Tableau Server boils down to these general approaches:

- **Optimize for User Traffic**: This tunes the server to respond to user requests and to display views quickly.

- **Optimize for Extracts**: This tunes the server to refresh extracts for published data sources. You might want to optimize for extract refreshes if your organization has a lot of data and the data needs to be as up to date as possible.

- **Optimize for Query Heavy Environments**: This is a specialized server configuration to optimize for query performance of workbooks that use extracts as their data source.

Rendering views and refreshing extracts generate the most load on the server, so you should optimize for the task that your organization is most interested in.

**Topics in this section:**

**Optimize for User Traffic**

You can optimize for traffic if you have many active Tableau Server users and few published data sources that need extract refreshes.

- When to optimize for user traffic

- **Ways to optimize for user traffic**

When to optimize for user traffic

Slow load times for views

Use the **Requests and Sessions** dashboard of the sample performance workbook to analyze how long views take to load.
If multiple views take longer than 10 seconds to load, and if the slow load times correspond to a large number of sessions, that can indicate that user traffic is slowing down the server.

However, if a particular view takes a long time to load regardless of when it is viewed, it’s a sign that the workbook for the view needs to be optimized. You can identify which workbooks need to be optimized with the Stats for Load Times administrative view. Some simple ways of optimizing workbooks includes displaying less information in each view or breaking up views, reducing the number of filters, and using data extracts.

High resource usage corresponding to user traffic

If your server displays high CPU and memory usage during peak traffic hours, you should optimize for user traffic. To determine peak traffic hours and analyze how many concurrent users are on your server, use the Users and Actions dashboard. In addition, you can use the Traffic to Views administrative view to see how much user traffic involves accessing views (as opposed to performing administrative functions, publishing, or other tasks).
If you click a point in the **Number of Users** view, the dashboard displays the users that were active at the time and the number of user actions that those users performed. By default, the only user actions displayed are user views, but you can use the **Action Types** filter to display additional user actions.

Make a note of the times of day when there are many concurrent users and views so that you can compare this to resource usage. As a rule of thumb, the number of users should correspond to a high number of user actions. However, the view in this example displays an artificially high number of actions for a single user as part of a load generation test. As an example, you can compare the high number of views at 12 AM on June 28th with the resource usage in the dashboard illustrated later.

Use the **CPU Usage** dashboard to display the percent of total CPU usage and the percent of CPU usage for each process. In the following example, note the large spike in total CPU usage and in the VizQL server process at 12 AM on June 28th. Because the VizQL server process loads and render views, the VizQL server process is often the first process to show strain under high user traffic.
**Note:** The percent of CPU usage for individual processes may add up to more than 100 percent. This is because processor utilization for individual processes is measured for a given processor core. By contrast, the total CPU usage is measured for all processor cores.

Use the **Memory Usage** dashboard to display the percent of total memory usage and the average memory usage in gigabytes. As a general rule, memory usage increases steadily with user traffic. Here again the VizQL server process is the first to show strain under high traffic.
Ways to optimize for user traffic

When high user traffic corresponds to high resource usage as it does in the example shown previously, you should optimize for user traffic.

Adjust the number of VizQL server processes

The most effective way of optimizing for user traffic is to adjust the number of VizQL server processes. Add one VizQL server process at a time and measure the effect with more performance monitoring. Because VizQL server processes can consume a lot of CPU and memory, adding too many processes can slow down the server instead. If you see consistently high memory usage, try to reduce the number of VizQL server processes to reduce the amount of memory reserved.

For more information about configuring processes, see Configure Nodes.

Adjust the number of other processes

Although the most effective way of improving performance for user traffic is to adjust the number of VizQL server processes, you can also tune other processes that support the VizQL server process or that prevent the VizQL server process from accessing resources.
For example, the VizQL server process makes frequent requests to the cache server process, so you might also want to increase the number of cache server processes. On the other hand, the Backgrounder processes might contend for CPU resources with the VizQL server process. As a result, if you do not need to run frequent extract refreshes, you might reduce the number of processes for the backgrounder. If you do need additional instances of the backgrounder, and if you’re running Tableau Server on a cluster, you can move the Backgrounder process to a dedicated node.

Adjust the VizQL session timeout limit

In the example shown previously, the amount of memory used by the VizQL server process increases with user traffic, and it remains reserved by Tableau Server for some time after the traffic finished. This is because the VizQL server process reserves memory for each session for a specified amount of time. If the VizQL server process uses a high percentage of the available memory, try reducing the timeout for each session to make memory available more quickly.

To do this, use the `tsm configuration set command to reduce the vizqlserver.session.expiry.timeout setting. The default is 30 minutes.`

Refresh the cache less often

If your users do not always need the most up-to-date data, you can optimize for user traffic by configuring Tableau Server to cache and reuse data as much as possible.

To do this, use the `tsm data-access caching list command to confirm the refresh frequency. The default is Low. Use the tsm data-access caching set command to change the refresh frequency.`

Assess view responsiveness

When a user opens a view, the components of the view are first retrieved and interpreted, then displayed in the user’s web browser. For most views, the display rendering phase occurs in the user’s web browser and in most cases, this yields the fastest results and highest level of interactive responsiveness. Handling most interactions in the client web browser reduces bandwidth and eliminates round-trip request latencies. If a view is very
complex, Tableau Server handles the rendering phase on the server instead of in the client web browser, because that generally results in the best performance. If you find that views aren't as responsive as you'd like, you can test and change the threshold that causes views to be rendered by the server instead of in the client web browser. For more information, see Configure Client-Side Rendering.

Configure Client-Side Rendering

When you navigate to a view in Tableau Server, the processing required to display the view, called rendering, can be performed by either your client device or Tableau Server. The choice depends on the complexity of the view, which is determined by the number of marks, rows, columns, and more. If a view is less complex, it's faster for a client device to render the view. If a view is more complex, it's faster to send a request to Tableau Server and take advantage of the server's greater computing power.

**Note:** If a view uses the polygon mark type or the page history feature, server-side rendering is always performed, even if client-side rendering is enabled.

Supported browsers

Client-side rendering is supported in Internet Explorer version 9.0 or higher, Firefox, Chrome, and Safari. All of these web browsers include the HTML 5 `<canvas>` element, which client-side rendering requires.

Client-side rendering is also supported by the Tableau Mobile app.

Configure the complexity threshold for computers and mobile devices

Because computers have more processing power than mobile devices, Tableau Server performs more client-side rendering on computers than on mobile devices.

As a server administrator, you can configure when client-side rendering happens on computers and mobile devices by adjusting the complexity threshold for each. For example, you
might lower the threshold for mobile devices if you notice that views display slowly on them. Or, you might increase the threshold to reduce the number of requests to Tableau Server.

By default, the complexity threshold for computer web browsers is 100. To adjust the complexity threshold for computers, use the following command:

```
tsm configuration set -k vizqlserver.browser.render_threshold -v [new value]
```

By default, the complexity threshold for mobile devices is 60. To adjust the complexity threshold for mobile devices, use the following command:

```
tsm configuration set -k vizqlserver.browser.render_threshold_mobile -v [new value]
```

For example, to change the mobile threshold to 40, you might enter the following command:

```
tsm configuration set -k vizqlserver.browser.render_threshold_mobile -v 40
```

For more information on how to use tsm option set, see tsm configuration set Options.

Disable client-side rendering

Client-side rendering is enabled by default and is recommended to improve the performance of views. However, you might want to disable client-side rendering temporarily for testing or if your server is being accessed primarily by computers or mobile devices with very little processing power.

Use the following command to disable client-side rendering:

```
tsm configuration set -k vizqlserver.browser.render -v false
```

For more information on how to use tsm option set, see tsm configuration set Options.
Testing with the URL Parameter

To test server-side rendering on a session basis, type `?:render=false` at the end of the view’s URL. For example:

http://localhost/views/Supplies/MyView?:render=false

If client-side rendering is disabled on Tableau Server, enter `?:render=true` to enable it for the session:

http://localhost/views/Supplies/MyView?:render=true

You can also test particular complexity thresholds on individual views to see if it’s appropriate to adjust the server-wide threshold for your server and network conditions. For example, you may find that lower complexity (such as 80) or higher complexity (such as 120) tipping points result in more responsiveness to user interactions. To test a threshold, you can keep the server’s default configuration (client-side-rendering enabled) and enter the test threshold number at the end of the view’s URL. For example:

http://localhost/views/Supplies/MyView?:render=80

Optimize for Extracts

Try to optimize for extracts if the extract schedules correspond to high resource usage or if extracts take a long time to finish.

When to optimize for extracts

High CPU usage corresponds to extract schedules

Use the **Background Jobs** dashboard of the sample performance workbook to view the number of background jobs run by Tableau Server, including extract refresh jobs. The dashboard also displays how long background jobs are delayed—that is, the amount of time between when a background job is scheduled and when it actually runs. If you see long delays at particular times of the day or if many jobs are running at the same time, try distributing the job schedules across different times of the day to reduce the load on the server.
Also compare the times when there are many background jobs or long delays with the CPU usage of the server. Use the CPU Usage dashboard to display the percent of total CPU usage and the percent of CPU usage for each process. Because the backgrounder process runs background jobs, it is the first process to show strain when there are many extract refresh jobs or when there are slow extract refresh jobs. Note that the CPU usage of the backgrounder process periodically but briefly reaches 100 percent. This indicates that there are intensive refresh jobs on a recurring schedule.

Note: The percent of CPU usage for individual processes may add up to more than 100 percent because processor utilization for individual processes is measured for a given processor core. By contrast, the total CPU usage is measured for all processor cores.
Extracts fail or run slowly

Use the Background Tasks for Extracts administrative view to determine how many extracts fail and how long extracts take to complete. Frequent failures can indicate a problem with a particular data source.
Ways to optimize for extracts

When high CPU usage corresponds to extract refresh schedules like it does in the example shown previously, you should optimize for extracts.

Adjust the extract refresh schedule

Use the **Background Jobs** dashboard of the sample performance workbook to identify optimal times for running extracts. In addition to running extracts in off-peak hours, you can distribute extract refreshes to minimize concurrent server load. If extract refreshes continue to cause problems, reduce the frequency of extract refreshes as much as possible in these ways:

- Schedule extracts for times when the server isn’t busy.
- Reduce the frequency of refreshes.
Speed up specific extracts

Use the Background Tasks for Extracts administrative view to identify failing extracts and long-running extracts.

- Reduce the size of extracts. You can help improve server performance by keeping the extract’s data set short, through filtering or aggregating, and narrow, by hiding unused fields. To make these changes, use the Tableau Desktop options Hide All Unused Fields and Aggregate data for visible dimensions. For more information, see Creating an Extract in the Tableau Help.

For general tips on building well-performing workbooks, search for “performance” in the Tableau Help. To see how workbooks perform after they’ve been published to Tableau Server, you can create a performance recording. For more information, see Create a Performance Recording.

- Use incremental refresh jobs. Incremental refresh jobs append new rows to an existing extract instead of creating the extract from scratch. This type of extract refresh runs quickly because it processes only the data that has been added since the last time the extract refresh job ran. However, it does not account for data that has been updated rather than appended to a data source. As a result, if you run incremental refresh jobs, you should still occasionally run full refresh jobs. For example, you might run a full refresh job once or twice a week for a data source instead of every day.

Configure the execution mode for extract refreshes

When you create extract refresh schedules, ensure that they run in parallel execution mode. When you run a schedule in parallel, it runs on all available backgrounder processes, even if the schedule contains only one refresh task. When you run a schedule serially, it only runs on one backgrounder process. By default, the execution mode is set to parallel so that refresh tasks finish as quickly as possible.

However, in some circumstances, it can make sense to set the execution mode to serial. For example, you might set the execution mode to serial if a very large job is preventing other schedules from running because it uses all available backgrounder processes.
Increase the number of backgrounder processes

A single background process can consume 100 percent of a single CPU core for certain tasks. As a result, the total number of instances you should run depends on the computer’s available cores. If you have Tableau Server installed in a cluster and you run backgrounder processes on a separate node, a good rule of thumb is to set the number of backgrounder process to between half the number of cores and the full number of cores of the computer running the backgrounder processes.

For more information about configuring processes, see Configure Nodes.

Isolate processes

If you have Tableau Server installed in a cluster, you see the largest benefit from moving the backgrounder processes to a separate node to avoid resource contention. This is because the backgrounder process is very CPU-intensive and running it on the same node where other CPU-intensive processes are running can slow down the server. For example, both the VizQL server process and the data engine process can be CPU-intensive. Read the two-node configuration in the Recommended Baseline Configurations topic for more details.

Optimize for Query Heavy Environments

This configuration has been tested by the Tableau team using Tableau Server Data Engine powered by Hyper technology and is recommended for Tableau Server versions 10.5 and later.

When to use this configuration

There are several factors that affect query performance, including, but not limited to: large extract size, complex calculations, or multiple views in a dashboard. Use this configuration to improve workbook performance if one of the two following conditions apply to your environment:

- Your workbooks use extracts and you are seeing long-running queries. In this context long-running queries means that the query execution time on extracts is taking more
You are seeing resource contention between Data Engine and VizQL Server when File Store is present on the box.

**Note:** Use performance recording to determine query execution times. To determine resource usage of Tableau use **Performance monitor** for Windows installations, and **sysstat** or **vmstat** tools for Linux installations.

**Important:** Network bandwidth is an important consideration for this configuration. Once a query result is generated, it is sent from the node with the File Store to the node with VizQL Server. Or when publishing an extract, the extract must be sent over the network to the File Store node. Both of these may take additional time depending on the size of data and network bandwidth available.

When not to use this configuration

If you are not seeing a large number of slow running queries (queries taking more than 5 seconds to complete), are unsure of your workload type, or do not see resource contention between Data Engine and VizQL Server when File Store is present, then this configuration may not be optimal for you. Before implementing this configuration, it is highly recommended to evaluate your CPU usage for VizQL Server and Data Engine installed with the File Store.

Furthermore, you should also consider the overall mix of workbooks in your environment. Tableau team tested this configuration using workbooks that have slow running queries (taking more than 5 seconds to complete) and workbooks with queries that take less than 5 seconds to complete. During that testing, Tableau team found that queries that take less than 5 seconds may get slightly slower. Depending on your workload mix of slow and fast running queries, you should evaluate whether this configuration will result in overall better performance for your environment.
Configuration

<table>
<thead>
<tr>
<th>Process</th>
<th>Initial Node</th>
<th>Additional Node 1</th>
<th>Additional Node 2</th>
<th>Additional Node 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cluster Controller</td>
<td>✔</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Gateway</td>
<td>✔</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Application Server</td>
<td>✔</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>VizQL Server</td>
<td>✔ ✔</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cache Server</td>
<td>✔ ✔</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Search &amp; Browse</td>
<td>✔</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Backgrounder</td>
<td>✔ ✔ ✔</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Data Server</td>
<td>✔ ✔</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Data Engine</td>
<td>✔</td>
<td></td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>File Store</td>
<td>✔ ✔</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Repository</td>
<td>✔ ✔</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The guiding principle for this configuration is to separate the node with File Store from the node that has VizQL Server installed. This is because of the following reasons:

- On Tableau Server, the Data Engine co-located with File Store is used for querying extracts.
- VizQL Server consumes noticeable resources when loading views and interactive use from a web browser. If both File Store and VizQL Server are on the same node, there can be contention for resources between VizQL Server and Data Engine, especially if you are running heavy queries (for example, workbooks with complex calculations) on extracts to load the views.

Thus, by separating VizQL Server and File Store processes, the load between querying extracts and loading or interacting with views can be balanced and better managed. This configuration is targeted at making slow queries faster. By slow queries, we mean queries that take more than 5 seconds to complete.

Here is a visual representation of how to configure your Tableau Server installation to optimize the performance of workbooks that use extracts and have long query execution times:
In this example, all computers are assumed to be at least 64-bit, 16 core, 16+GB systems.

General configuration notes

**These configuration guidelines apply to all configurations for Tableau Server installations.**

- To calculate the minimum number of Backgrounder processes to run, divide the computer’s total number of physical cores by 4. To calculate the maximum number, divide the computer’s total physical cores by 2.

- Run two VizQL Server processes, two Cache Server processes, and two Data Server processes on the nodes that are not running File Store processes.

Special configuration notes

**These configuration recommendations are specific to this configuration.**

- For this configuration, typically 16 cores are recommended for each node, with 16 GB of RAM per core. Adding more cores to the node where File Store is running will improve performance since Data Engine is designed to maximize and leverage available cores on the machine. In tests performed by the Tableau team indicates that the File Store/Data Engine dedicated nodes benefited from adding more cores. Pursuing this configuration will allow for more flexibility when scaling your deployment as you are now able to scale the File Store node independently of other processes.

- Run one process of File Store on a separate node that is not running VizQL or Backgrounder processes. Tableau also strongly recommends that you compare the CPU usage for Data Engine to the CPU usage for VizQL Server, and distribute the cores for each of the processes accordingly. The exact ratio can vary depending on your workload type, but a starting point would be to have one File Store/Data Engine node for every 2 VizQL Server nodes.

- The default of this configuration is 4 nodes for high availability considerations.
Performance Benefits

Here are the general benefits you can expect to see from this configuration and the trade-offs you should consider when adopting this configuration:

<table>
<thead>
<tr>
<th>Benefits</th>
<th>Trade-offs</th>
</tr>
</thead>
<tbody>
<tr>
<td>More consistent performance in user experience when loading extract-dependent workbooks*</td>
<td>Fast-running queries may potentially get slightly slower</td>
</tr>
<tr>
<td>Better response times on workbooks with long-running queries*</td>
<td>You may have fewer VizQL Server processes moving to this topology without growing your deployment</td>
</tr>
<tr>
<td>More scalability options to grow your deployment</td>
<td></td>
</tr>
<tr>
<td>Reduction in resource consumption on nodes with VizQL Server</td>
<td></td>
</tr>
<tr>
<td>Reduction in CPU spikes above 95%</td>
<td></td>
</tr>
</tbody>
</table>

*While under moderate (<80%) CPU usage

Additional Benefits

- In the tests done by the Tableau team, we found that for the same number of users, the more the number of query intensive workbooks (which use extracts) in an environment, the faster the time it took to reach the saturation point. If you want to make these workbooks more performant, or service more users using these workbooks, this configuration will help you achieve these objectives.

- This configuration also opens a new scalability option for your installation. You now have the option to scale horizontally or vertically with respect to querying on extracts. Unlike other Tableau processes, the Data Engine leverages as many cores as available on the machine. Given this, you have the flexibility to add more resources to the
standalone File Store nodes to reduce query response time and variability on expensive extract queries or add more File Store nodes to get more extract query throughput in your Server.

When to Add Nodes and Reconfigure

Tableau Server can scale up and out as your needs and requirements evolve. Here are some guidelines to help you figure out whether it’s time to add more nodes to your system, reconfigure the server, or both:

- **More than 100 concurrent users**: If your deployment is user-intensive (>100 simultaneous viewers), it’s important to have enough VisQL processes—but not so many that they exceed your hardware’s capacity to handle them. Also, enabling the Tableau Server Guest User account can increase the number of potential simultaneous viewers beyond the user list you may think you have. The administrative view can help you gauge this. For more information, see Actions by Specific User.

- **Heavy use of extracts and frequent extract refreshes**: Extracts can consume a lot of memory and CPU resources. There’s no one measurement that qualifies a site as extract-intensive. Having just a few, extremely large extracts could put your site in this category, as would having very many small extracts. Sites where extracts are frequently refreshed (for example, several times a day) are often helped by more emphasis on the background process, which handles refresh tasks. Use the Background Tasks for Extracts administrative view to see your current refresh rate. Extract heavy sites benefit from isolating the Backgrounder process on its own machine. For more information, see the two-node configuration in the Recommended Baseline Configurations topic.

- **Query heavy environments**: If you are experiencing slow query performance for Workbooks that use extracts, isolating nodes that handle queries on extracts from VisQL processes can improve and stabilize performance. For more information, see Optimize for Query Heavy Environments.

- **Downtime potential**: If your server system is considered mission critical and requires
a high level of availability, you can configure it so there’s redundancy for the server processes that handle extracts, the repository, and the gateway.

Performance Tuning Examples

Performance tuning examples have been expanded and moved to separate topics. For more information, see Performance Tuning.

For recommended baseline configurations that serve as starting point for installations with one, two, and three nodes, see Recommended Baseline Configurations.

Performance Recording

This section describes how to create performance recordings and use the results to improve workbook performance. With performance recordings, you can view how long workbook events take. For example, you can see how long it takes to connect to a data source, run a query, render data, and more.

Create a Performance Recording

The Performance Recording feature in Tableau records performance information about key events as you interact with a workbook. You can then view performance metrics in a workbook that Tableau creates to analyze and troubleshoot different events that are known to affect performance:

- Query execution
- Geocoding
- Connections to data sources
- Layout computations
- Extract generation
Tableau Server on Linux Administrator Guide

- Blending data
- Server blending (Tableau Server only)

Tableau support may ask that you create a performance workbook as they work with you to diagnose performance issues.

Looking for Tableau Server on Windows? See Create a Performance Recording.

Enable Performance Recording for a Site

By default, performance recording is not enabled for a site. A server administrator can enable performance recording site by site.

1. Navigate to the site for which you want to enable performance recording.

2. Click Settings:


4. Click Save.

Start a Performance Recording for a View

1. Open the view for which you want to record performance.

   When you open a view, Tableau Server appends ":iid=<n>" after the URL. This is a session ID. For example:
2. Type `:record_performance=yes` at the end of the view URL, immediately before the session ID. For example:

   http://10.32.139.22/#/views/Coffee_Sales2013/USSalesMarginsByAreaCode?:iid=1

3. Click the **Refresh** button in the toolbar.

4. Load the view.

   A visual confirmation that performance recording has started is the **Performance** option in the view toolbar:

![Performance option in toolbar](image)

**View a Performance Recording**

1. Click **Performance** to open a performance workbook. This is an up-to-the-minute snapshot of performance data. You can continue taking additional snapshots as you continue working with the view; the performance data is cumulative.

2. Move to a different page or remove `:record_performance=yes` from the URL to stop recording.

**Interpret a Performance Recording**

A performance recording workbook is a Tableau dashboard that contains three views: **Timeline**, **Events**, and **Query**.
For information on how to create a performance recording in Tableau Server, see Create a Performance Recording.

Timeline

The uppermost view in a performance recording dashboard shows the events that occurred during recording, arranged chronologically from left to right. The bottom axis shows elapsed time since Tableau started, in seconds.

In the Timeline view, the Workbook, Dashboard, and Worksheet columns identify the context for events. The Event column identifies the nature of the event, and the final column shows each event's duration and how it compares chronologically to other recorded events.

Events

The middle view in a performance recording workbook shows the events, sorted by duration (greatest to least). Events with longer durations can help you identify where to look first if you want to speed up your workbook.
Different colors indicate different types of events. The range of events that can be recorded is:

- Computing layouts
  
  If layouts are taking too long, consider simplifying your workbook.

- Connecting to data source
  
  Slow connections could be due to network issues or issues with the database server.

- Executing query
  
  - For live connections, if queries are taking too long, it could be because the underlying data structure isn’t optimized for Tableau. Consult your database server’s documentation. As an alternative, consider using an extract to speed performance.
  
  - For extracts, if queries are taking too long, review your use of filters. If you have a lot of filters, would a context filter make more sense? If you have a dashboard that uses filters, consider using action filters, which can help with performance.

- Generating extract
  
  To speed up extract generation, consider only importing some data from the original data source. For example, you can filter on specific data fields, or create a sample based on a specified number of rows or percentage of the data.

- Geocoding
  
  To speed up geocoding performance, try using less data or filtering out data.

- Blending data
  
  To speed up data blending, try using less data or filtering out data.

- Server rendering
You can speed up server rendering by running additional VizQL Server processes on additional machines.

Query

If you click on an **Executing Query** event in either the **Timeline** or **Events** section of a performance recording dashboard, the text for that query is displayed in the Query section.

If you are connected to a published data source, the query text is displayed in XML. If you are connected to the data source directly, the query is displayed in SQL like shown below:

```sql
SELECT "State"."ID" AS "ID",
"StateSynonyms"."Name" AS "State_Name",
"State"."ParentID" AS "State_ParentID"
FROM "StateSynonyms"
INNER JOIN "State" ON ("State"."ID" = "StateSynonyms"."ParentID")
AND ("State"."MapCode" = "StateSynonyms"."MapCode")
```

If it makes sense, you can use the query text to work with your database team on optimizing at the database level. Sometimes the query is truncated and you’ll need to look in the Tableau log to find the full query. Most database servers can give you advice about how to optimize a query by adding indexes or other techniques. See your database server documentation for details.

Sometimes for efficiency, Tableau combines multiple queries into a single query against the data. In this case, you may see an **Executing Query** event for the Null worksheet and zero queries being executed for your named worksheets.

**Performance Monitoring Tools**

This topic describes external resources that you can use to monitor and tune performance.

**Disclaimer**: This topic includes information about third-party and community supported products. Please note that while we make every effort to keep references to third-party and community content accurate, the information we provide here might change without
notice. For the most up-to-date information, please consult the documentation for products referenced here. To learn more about community supported tools, see Support levels for IT and developer tools.

- **TabJolt.** A load and performance testing tool that you can use to understand how Tableau Server scales with your workloads, in your environment, and to inform your scalability and capacity needs. Here are some key use cases for when you would use TabJolt:
  - To establish a baseline for server performance and test deployments before pushing them to production environments.
  - In a new Tableau Server, to help understand how the new server scales in your environment, specifically to your hardware and workload?
  - Before upgrading to understand the new version will scale in your environment.
  - To find the best server deployment configuration, given your hardware, workbooks and environments.

- **Replayer.** A tool that can replay log-based real user traffic from a Tableau Server against any other server or configuration. It replays Tableau Server single- or multi-user sessions. Here are some ways that Replay can be used:
  - Playback specific Tableau Server sessions, and filter the session based upon start time or RequestID.
  - Use it to simulate load conditions so that you can test how to scale and balance your Tableau Server installations.
  - Perform regression testing by running and comparing end-to-end user scenarios for Tableau Server upgrades.
Capture and report HTTP exceptions that occur in a single-user session.

Replay a defect, so that you can troubleshoot and verify that it is fixed.

**Scout.** An exploratory tool that captures performance metrics across any workbooks on both Tableau Desktop and Tableau Server. Here are some ways that Scout can be used:

- Find slow workbooks on Server.
- Validate performance improvements or regressions after making server configuration or topology changes.
- Validate that workbooks and dashboards are loading properly after upgrading to new Tableau Server.
- Validate that workbooks are still working properly after data source changes.

**Sitescope.** An agentless application monitoring tool.

**Zabbix.** An open-source, real-time monitoring tool.

**Splunk.** A tool for monitoring and analyzing machine data, including logs.

**Graylog.** An open-source log management tool.

**Configure Client-Side Rendering**

When you navigate to a view in Tableau Server, the processing required to display the view, called rendering, can be performed by either your client device or Tableau Server. The choice depends on the complexity of the view, which is determined by the number of marks, rows, columns, and more. If a view is less complex, it's faster for a client device to render the view. If a view is more complex, it's faster to send a request to Tableau Server and take advantage of the server’s greater computing power.
Note: If a view uses the polygon mark type or the page history feature, server-side rendering is always performed, even if client-side rendering is enabled.

Supported browsers

Client-side rendering is supported in Internet Explorer version 9.0 or higher, Firefox, Chrome, and Safari. All of these web browsers include the HTML 5 `<canvas>` element, which client-side rendering requires.

Client-side rendering is also supported by the Tableau Mobile app.

Configure the complexity threshold for computers and mobile devices

Because computers have more processing power than mobile devices, Tableau Server performs more client-side rendering on computers than on mobile devices.

As a server administrator, you can configure when client-side rendering happens on computers and mobile devices by adjusting the complexity threshold for each. For example, you might lower the threshold for mobile devices if you notice that views display slowly on them. Or, you might increase the threshold to reduce the number of requests to Tableau Server.

By default, the complexity threshold for computer web browsers is 100. To adjust the complexity threshold for computers, use the following command:

```
tsm configuration set -k vizqlserver.browser.render_threshold -v [new value]
```

By default, the complexity threshold for mobile devices is 60. To adjust the complexity threshold for mobile devices, use the following command:

```
tsm configuration set -k vizqlserver.browser.render_threshold_mobile -v [new value]
```

For example, to change the mobile threshold to 40, you might enter the following command:
Tableau Server on Linux Administrator Guide

```bash
tsm configuration set -k vizqlserver.browser.render_threshold_mobile -v 40
```

For more information on how to use tsm option set, see tsm configuration set Options.

**Disable client-side rendering**

Client-side rendering is enabled by default and is recommended to improve the performance of views. However, you might want to disable client-side rendering temporarily for testing or if your server is being accessed primarily by computers or mobile devices with very little processing power.

Use the following command to disable client-side rendering:

```bash
tsm configuration set -k vizqlserver.browser.render -v false
```

For more information on how to use tsm option set, see tsm configuration set Options.

**Testing with the URL Parameter**

To test server-side rendering on a session basis, type `?:render=false` at the end of the view’s URL. For example:

```
http://localhost/views/Supplies/MyView?:render=false
```

If client-side rendering is disabled on Tableau Server, enter `?:render=true` to enable it for the session:

```
http://localhost/views/Supplies/MyView?:render=true
```

You can also test particular complexity thresholds on individual views to see if it’s appropriate to adjust the server-wide threshold for your server and network conditions. For example, you may find that lower complexity (such as 80) or higher complexity (such as 120) tipping points result in more responsiveness to user interactions. To test a threshold, you can keep the server’s default configuration (client-side-rendering enabled) and enter the test threshold number at the end of the view’s URL. For example:

```
http://localhost/views/Supplies/MyView?:render=80
```
Monitoring Tableau Server

You can configure SMTP and alerts and subscriptions to aid in monitoring Tableau Server.

Configure SMTP Setup

Tableau Server can email server administrators about system failures, and email server users about subscribed views and data-driven alerts. First, however, you need to configure the SMTP server that Tableau Server uses to send email. After configuring SMTP, complete the steps to configure notifications (Configure Server Event Notification), then when you start or restart the server, it will trigger an email notification, which confirms that you have set up notifications correctly.

Configuring SMTP requires that you restart Tableau Server services.

Secure SMTP

To enable and configure TLS for SMTP, you must use the TSM CLI as described in this topic.

If your organization does not use public certificates for verifying TLS connections, then you can upload a private certificate to Tableau Server to verify trusted connections. For more information, see the tsm security custom-cert add command.

You may also configure SMTP TLS for encryption only by disabling the certificate validation process. See the section, Configuration file reference, later in this topic for more information.

Use the TSM web interface

1. Open TSM in a browser:

   https://<tsm-computer-name>:8850. For more information, see Sign in to Tableau Services Manager Web UI.

2. Click Notifications on the Configuration tab and click Email Server.
3. Enter the SMTP configuration information for your organization:

   ![SMTP Configuration Screen](image)

4. Click **Save Pending Changes** after you've entered your configuration information.

5. Click **Pending Changes** at the top of the page:

   ![Pending Changes](image)

6. Click **Apply Changes and Restart**.

7. Run the `tsm email test-smtp-connection` to view and verify the connection configuration. See `tsm email test-smtp-connection`.

**Use the TSM CLI**

For the initial configuration of SMTP, we recommend that you use the configuration file template below to create a JSON file. You can also set any single configuration key listed below with the syntax described in `tsm configuration set`. 
1. Copy the following json template to a file.

```
{
  "configKeys": {
    "svcmonitor.notification.smtp.server": "SMTP server host name",
    "svcmonitor.notification.smtp.send_account": "SMTP user name",
    "svcmonitor.notification.smtp.port": 443,
    "svcmonitor.notification.smtp.password": "SMTP server password",
    "svcmonitor.notification.smtp.ssl_enabled": true,
    "svcmonitor.notification.smtp.from_address": "From email address",
    "svcmonitor.notification.smtp.target_addresses": "To email address1,address2",
    "svcmonitor.notification.smtp.canonical_url": "Tableau Server URL"
  }
}
```

**Important:** The template below include common options for most deployments. After you copy the template to a text file, you must edit the option values for your SMTP server requirements. You may need to remove or add options. See the reference section that follows for more information about all supported SMTP key options.

2. Run the `tsm settings import -f file.json` to pass the json file with the
appropriate values to Tableau Services Manager to configure Tableau Server for SMTP. Tableau Services Manager will validate the entity values.

3. Run the `tsm pending-changes apply` command to apply the changes. See `tsm pending-changes apply`.

4. Run the `tsm email test-smtp-connection` to view and verify the connection configuration. See `tsm email test-smtp-connection`.

Configuration file reference

This table lists all of the options that can be used to configure SMTP.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>svc-mon-itor.notification.smtp.server</code></td>
<td>Address of SMTP server. Example:</td>
</tr>
<tr>
<td></td>
<td>&quot;svc-mon-itor.notification.smtp.server&quot;:</td>
</tr>
<tr>
<td></td>
<td>&quot;mail.example.com&quot;</td>
</tr>
<tr>
<td><code>svc-mon-itor.notification.smtp.send_account</code></td>
<td>User name for SMTP account.</td>
</tr>
<tr>
<td><code>svc-mon-itor.notification.smtp.port</code></td>
<td>Port number for SMTP server. The default is 25.</td>
</tr>
<tr>
<td><code>svc-mon-itor.notification.smtp.password</code></td>
<td>Password for SMTP server account. Example:</td>
</tr>
<tr>
<td></td>
<td>&quot;svc-&quot;</td>
</tr>
<tr>
<td>Option</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------------------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>monitor.notification.smtp.password&quot;::&quot;password</td>
<td>Specifies whether the connection to the SMTP server is encrypted. The default is false.</td>
</tr>
<tr>
<td>svc-monitor.notification.smtp.ssl_enabled</td>
<td>If enabled, Tableau Server will refuse to connect to SMTP servers without using TLS The <code>svc-monitor.notification.smtp.ssl_enabled</code> option must also be set to true. The default is false.</td>
</tr>
<tr>
<td>svc-monitor.notification.smtp.ssl_required</td>
<td>If set to true, Tableau Server will check the SMTP server identity as specified by RFC 2595. These additional checks based on the content of the server's certificate are intended to prevent man-in-the-middle attacks. The default is false.</td>
</tr>
<tr>
<td>svc-monitor.notification.smtp.ssl_check_server_identity</td>
<td>When using TLS, trust certificates from all mail servers, ignoring the validity of the certificate's chain of trust. By setting this key to true, TLS will be used only to encrypt the traffic to the SMTP host. The default is false.</td>
</tr>
<tr>
<td>svc-monitor.notification.smtp.ssl_trust_all_hosts</td>
<td>The default and supported sets of cipher suites is defined by the version of JDK that is installed.</td>
</tr>
<tr>
<td>Option</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td><code>itor.no-tification.smtp.ssl_cipher</code></td>
<td>with Tableau Server. See the section below, TLS ciphers, for a list of supported and default ciphers. To update the cipher suites used by Tableau Server for SMTP TLS connections, enter a white space-separated list of cipher suites for this value. For example, &quot;TLS_ECDH_ECDSA_WITH_AES_128_CBC_SHA256 TLS_DHE_DSS_WITH_AES_128_GCM_SHA256 TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA384&quot;.</td>
</tr>
<tr>
<td><code>svc-monitor-itor.notification.smtp.ssl_versions</code></td>
<td>The default TLS versions enabled on this version of Tableau Server are TLSv1, TLSv1.1, TLSv1.2. TLS version support is defined by the version of JDK that is installed with Tableau Server. Supported versions of TLS are SSLv2Hello, SSLv3, TLSv1, TLSv1.1, TLSv1.2. To update the versions used by Tableau Server for SMTP TLS connections, enter a white space-separated list of versions for this value. For example, &quot;TLSv1.1 TLSv1.2&quot;.</td>
</tr>
<tr>
<td><code>svc-monitor-itor.notification.smtp.from_address</code></td>
<td>Email address that will send an notification if there’s a system failure. The email address must have valid syntax (for example, <a href="mailto:ITalerts@bigco.com">ITalerts@bigco.com</a> or noreply@mycompany), but it does not have to be an actual email account on Tableau</td>
</tr>
<tr>
<td>Option</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
|                                            | Server. (Some SMTP servers may require an actual email account, however.)  

**Note:** You can override the system-wide email address on a per-site basis. For more information, see What is a site.  

Example:  

"svc-monitor.notification.smtp.from_address": "donot-reply@example.com"

| svc-monitor.notification.smtp.target_addresses | Email address to receive notifications. If email notifications are enabled, you need to include at least one address. Separate multiple addresses with commas.  

Example:  

"svc-monitor.notification.smtp.target_addresses": "iluvdata@example.com"

| svc-monitor.notification.smtp.canonical_url | URL of the Tableau Server. Enter http:// or https://, followed by the name or IP address of the Tableau server. Used in the footer of subscription email.  

Example: |
Tableau Server on Linux Administrator Guide

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>&quot;svc-monitor.notification.smtp.canonical_url&quot;:</td>
<td>&quot;<a href="http://myserver.example.com">http://myserver.example.com</a>&quot;</td>
</tr>
</tbody>
</table>

### TLS ciphers

The following is a list of TLS ciphers that are supported by the JDK that is included with Tableau Server 2019.4. In this version of Tableau Server, all of these ciphers are enabled by default. You can specify a custom list of ciphers for your SMTP configuration by entering a white-space separated list with the option, svcmonitor.notification.smtp.ssl_ciphers, as described in the table above.

<table>
<thead>
<tr>
<th>TLS_RSA_WITH_AES_128_CBC_SHA256</th>
<th>TLS_ECDH_RSA_WITH_AES_256_GCM_SHA384</th>
</tr>
</thead>
<tbody>
<tr>
<td>TLS_DHE_DSS_WITH_AES_256_GCM_SHA384</td>
<td>TLS_ECDH_RSA_WITH_AES_128_GCM_SHA256</td>
</tr>
<tr>
<td>TLS_ECDH_RSA_WITH_AES_256_CBC_SHA384</td>
<td>TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA</td>
</tr>
<tr>
<td>TLS_ECDHE_ECDSA_WITH_AES_128_CBC_SHA</td>
<td>TLS_RSA_WITH_AES_256_CBC_SHA256</td>
</tr>
<tr>
<td>TLS_RSA_WITH_AES_128_GCM_SHA256</td>
<td>TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA384</td>
</tr>
<tr>
<td>TLS_ECDH_RSA_WITH_AES_128_CBC_SHA256</td>
<td>TLS_ECDHE_ECDSA_WITH_AES_128_GCM_SHA256</td>
</tr>
<tr>
<td>TLS_DHE_DSS_WITH_AES_128_CBC_SHA</td>
<td>TLS_DHE_DSS_WITH_AES_256_CBC_SHA256</td>
</tr>
<tr>
<td>SHA256</td>
<td>SHA</td>
</tr>
<tr>
<td>--------</td>
<td>-----</td>
</tr>
<tr>
<td>TLS_DHE_RSA_WITH_AES_128_CBC_SHA256</td>
<td>TLS_DHE_RSA_WITH_AES_256_CBC_SHA256</td>
</tr>
<tr>
<td>TLS_RSA_WITH_AES_256_GCM_SHA384</td>
<td>TLS_EMPTY_RENEGOTIATION_INFO_SCSV</td>
</tr>
<tr>
<td>TLS_ECDH_ECDSA_WITH_AES_256_CBC_SHA256</td>
<td>TLS_ECDHE_ECDSA_WITH_AES_256_CBC_SHA256</td>
</tr>
<tr>
<td>TLS_RSA_WITH_AES_128_CBC_SHA</td>
<td>TLS_DHE_DSS_WITH_AES_256_CBC_SHA256</td>
</tr>
<tr>
<td>TLS_RSA_WITH_AES_256_CBC_SHA</td>
<td>TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256</td>
</tr>
<tr>
<td>TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256</td>
<td>TLS_DHE_RSA_WITH_AES_256_CBC_SHA</td>
</tr>
<tr>
<td>TLS_DHE_DSS_WITH_AES_128_GCM_SHA256</td>
<td>TLS_ECDHE_RSA_WITH_AES_128_GCM_SHA256</td>
</tr>
<tr>
<td>TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA256</td>
<td>TLS_ECDH_RSA_WITH_AES_128_CBC_SHA</td>
</tr>
<tr>
<td>TLS_DHE_RSA_WITH_AES_256_GCM_SHA384</td>
<td>TLS_DHE_RSA_WITH_AES_128_CBC_SHA</td>
</tr>
<tr>
<td>TLS_ECDHE_RSA_WITH_AES_128_CBC_SHA256</td>
<td>TLS_ECDHE_RSA_WITH_AES_256_CBC_SHA</td>
</tr>
<tr>
<td>TLS_DHE_RSA_WITH_AES_128_GCM_SHA256</td>
<td>TLS_ECDH_ECDSA_WITH_AES_128_GCM_SHA256</td>
</tr>
<tr>
<td>TLS_ECDH_RSA_WITH_AES_256_CBC_SHA256</td>
<td>TLS_DHE_DSS_WITH_AES_128_CBC_SHA</td>
</tr>
</tbody>
</table>
Configure Server Event Notification

A Tableau Services Manager (TSM) administrator can configure Tableau Server to allow notifications for the following events:

- **Content updates**
  - Extract failures (enabled by default)
  - Subscription views for users (disabled by default)
- **Server health monitoring**
  - Server status changes (disabled by default)
  - License reporting (disabled by default)
- **Drive space**
  - Email alerts when space crosses or remains below pre-configured thresholds (disabled by default)
  - Recording usage history (enabled by default)

**Note:** You need to configure SMTP before you can configure subscriptions or notifications. For more information, see Configure SMTP Setup.

Use the TSM web interface

1. Open TSM in a browser:

   https://<tsm-computer-name>:8850.
For more information, see Sign in to Tableau Services Manager Web UI.

2. Click **Notifications** on the **Configuration** tab and click **Events**.

3. Configure notification settings for your organization:
   - Content updates
      - **Send emails for extract refresh failures**
        When this option is enabled (the default), a server administrator can configure email notifications to be sent when extract refreshes fail. These messages are configured at the site level, so even if this option is enabled, messages are not sent unless the **Send email to data source and workbook owners when scheduled refreshes fail** option is enabled for a site (this is enabled by default). For details, see Enable Extract Refresh Scheduling and Failure Notification.

      - **Allow users to receive email for views that they have subscribed to**
        When this option is enabled (by default is it disabled), a server administrator can configure a site to send subscription email. These email messages are configured at the site level and can only be configured when this option is enabled. For details, see Set Up a Site for Subscriptions.
        
        When users subscribe to a workbook or view, a snapshot of the view is emailed to them on a scheduled basis, so they can see the latest updates without having to sign into Tableau Server.
        
        To allow users to attach PDF renderings on subscription emails, select **Let users add attachments to subscribed views**.
Tableau Server on Linux Administrator Guide

- Server health monitoring

  - **Send emails for Tableau Server process events (up, down, and failover)**

    Tableau Server sends an email message when the data engine, repository, or gateway server processes stop or restart, or when the initial Tableau Server node stops or restarts.

    If you are running a single-server installation (all processes on the same computer), health alerts are only sent when Tableau Server is up. No "down" alerts are sent. If you are running a distributed installation that's configured for failover, a DOWN alert means that the active repository or a data engine instance has failed and the subsequent UP alert means that the passive instance (repository) or second instance (data engine) of that process has taken over.

  - **Send emails for Tableau Server license reporting**

    License reporting is generated at Tableau Desktop and sent to Tableau Server.

- Drive space

  Enable notifications for remaining drive space on your Tableau Server.

  - **Send emails when unused drive space drops below thresholds**

    You can configure Tableau Server to send email alerts when disk space usage on any node crosses a threshold, or remains below the threshold. And you can configure how often threshold alerts are sent.
There are two thresholds you must set, **Warning threshold** and **Critical threshold**. Thresholds are expressed in percentage of disk space remaining. The critical threshold must be less than the warning threshold.

You also specify the **Send threshold alert every** option. This determines how often, in minutes, warning and critical alerts should be sent. The default value is 60 minutes.

- **Record disk space usage information and threshold violations for use in custom administrative views**

  When you configure Tableau Server to record disk space usage, information about free disk space is saved in the repository and you can view the usage history using the Administrative Views.

4. Click **Save Pending Changes** after you’ve entered your configuration information.

5. Click **Pending Changes** at the top of the page:

6. Click **Apply Changes and Restart**.

**Use the TSM CLI**

The various notification values described above can be set individually with the tsm configuration set command. Alternatively, you can construct a json file and pass all configuration values in one operation. Both methods are described in this section.

Set notification values individually

The following table shows the key/value pairs that map to the notification events described earlier in this topic. Use the tsm configuration set command with the following syntax to set a single key/value pair:
Tableau Server on Linux Administrator Guide

```bash
  tsm configuration set -k <config.key> -v <config_value>
```

For example, to enable job failure notifications, run the following command:

```bash
  tsm configuration set -k backgrounder.notifications_enabled -v true
```

<table>
<thead>
<tr>
<th>Notification option</th>
<th>Key</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Extract failures or Flow run failures</td>
<td>backgrounder.notifications_enabled</td>
<td>true</td>
</tr>
<tr>
<td>Enable subscription views for user</td>
<td>subscriptions.enabled</td>
<td>true</td>
</tr>
<tr>
<td>Enable PDF attachments for subscriptions</td>
<td>subscriptions.attachments_enabled</td>
<td>true</td>
</tr>
<tr>
<td>Maximum attachment size (MB) for subscription notifications</td>
<td>subscriptions.max_attachment_size_megabytes</td>
<td>integer value, default is 150</td>
</tr>
<tr>
<td>Server status changes</td>
<td>svcmonitor.notification.smtp.enabled</td>
<td>true</td>
</tr>
<tr>
<td>License reporting</td>
<td>features.DesktopReporting</td>
<td>true</td>
</tr>
<tr>
<td>Remaining space thresholds: enable email</td>
<td>storage.monitoring.email_enabled</td>
<td>true</td>
</tr>
<tr>
<td>notifications</td>
<td>Remaining space thresholds: warning percentage</td>
<td>storage.monitoring.warning_percent</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>-----------------------------------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td></td>
<td>Remaining space thresholds: critical percentage</td>
<td>storage.monitoring.critical_percent</td>
</tr>
<tr>
<td></td>
<td>Set email interval</td>
<td>storage.monitoring.email_interval_min</td>
</tr>
<tr>
<td></td>
<td>Record usage history</td>
<td>storage.monitoring.record_history_enabled</td>
</tr>
</tbody>
</table>

After you are done setting values, you must run the following command:

```bash
tsm pending-changes apply
```

The `pending-changes apply` command displays a prompt to let you know this will restart Tableau Server if the server is running. The prompt displays even if the server is stopped, but in that case there is no restart. You can suppress the prompt using the `--ignore-prompt` option, but this does not change the restart behavior. For more information, see `tsm pending-changes apply`.

Set all notification values with a single json file

To make all notifications settings with a single configuration, you can pass a json file.

Copy and edit the following template to create a file for your configuration.

```json
{
    "configKeys": {
```
"backgrounder.notifications_enabled": true,
"subscriptions.enabled": true,
"subscriptions.attachments_enabled": true,
"subscriptions.max_attachment_size_megabytes": 150,
"svcmonitor.notification.smtp.enabled": true,
"features.DesktopReporting": true,
"storage.monitoring.email_enabled": true,
"storage.monitoring.warning_percent": 20,
"storage.monitoring.critical_percent": 15,
"storage.monitoring.email_interval_min": 25,
"storage.monitoring.record_history_enabled": true
}
}

After you have saved the file, pass it with the following command:

tsm settings import -f <path-to-file.json>

To apply changes, run the following command:

tsm pending-changes apply

The pending-changes apply command displays a prompt to let you know this will restart Tableau Server if the server is running. The prompt displays even if the server is stopped, but in that case there is no restart. You can suppress the prompt using the --ignore-prompt option, but this does not change the restart behavior. For more information, see tsm pending-changes apply.

Maintenance

You should perform regular maintenance on Tableau Server, such as creating backups, synchronizing Active Directory groups, and removing unneeded files.
Backup and Restore

A Tableau Server administrator should perform regular database maintenance, monitor disk usage on the server, clean up unnecessary files to free up space on the server, and back up Tableau Server and its data. Taking these steps can help ensure that Tableau Server runs with maximum efficiency.

You can use the Tableau Services Manager (TSM) command line tool to back up and restore your Tableau data. Tableau data includes data extract files, as well as Tableau Server’s own PostgreSQL database, which stores workbook and user metadata, and server configuration data. Tableau Server log files capture activity and can help you diagnose problems. Logs are written to folders on the server and you can archive and remove them to save disk space.

**Note:** You can use the `tsm maintenance restore` command to restore Tableau Server backups created using `tabadmin backup` and `tsm maintenance backup`. Database backups made in other ways, and virtual machine snapshots are not valid sources for restoring Tableau Server.

You can only restore from a backup that has the same type of identity store as the running server. For example, a backup from a server using local authentication can be restored to a Tableau Server initialized with local authentication, but a backup from a server using Active Directory authentication cannot be restored to a server initialized with local authentication.

Platform compatibility

You can use a backup created with Tableau Server on Linux to restore Tableau Server on Windows version 2018.2 and later.

You cannot use a backup created with Tableau Server on Linux to restore earlier versions of Tableau Server on Windows (version 2018.1 and earlier).
You can use a backup created with Tableau Server on Windows (version 2018.2 and earlier) to restore Tableau Server on Linux. For more information about restoring a Windows backup on Tableau Server on Linux, see Migrate Tableau Server from Windows to Linux.

Disk Space Usage for Backup and Restore

Backup disk space requirements

You need sufficient free disk space to restore from backup, that is in all cases 15% of the disk size. In some cases it could be even greater. During backups, the background tasks for cleaning up old extracts are temporarily paused. This means that, for the duration of the backup, extract refreshes will leave extra files in place, adding to disk space usage. If your backup takes a long time, or if your organization uses many extracts that are regularly updated, this can result in a significant amount of temporary disk space usage. These old files will be removed after the backup is complete.

- On nodes with the repository, you need additional free space equal to at least twice the size of the repository data. To obtain an estimate of the repository data, check the size of <data directory>/pgsql/data/base directory.

- On nodes with the file store, you need additional free space equal to at least the size of the file store data. To obtain an estimate of file store data (extracts, flows, etc.), check the size of <data directory>/dataengine directory.

- On nodes with the Administration Controller or TSM Controller (usually the initial node unless you have had to recover from an initial node failure), you need additional free space equal to at least the size of the repository data plus the size of the file store data.

If all of the above are on the same node, you can use the sum of the sizes to determine the amount of disk space required.

Restore disk space requirements

You must have adequate disk space for the database restore process to run successfully.

To restore Tableau Server:
On controller nodes, you need free space equal to at least the size of the backup archive.

On repository nodes, you need free space equal to at least twice the size of the repository data in the backup archive.

On file store nodes, you need free space equal to at least twice the size of the data-engine folder in the backup archive.

**Best Practices for Backing Up Tableau Server**

There are several ways you can maximize backup efficiency. Your environment can impact how effective each of these is, so test with your data to see what works best.

- Locating the repository (pgsql) and the File Store on the same node as the Administration Controller can reduce the length of time it takes to back up Tableau Server. The Administration Controller is usually on the initial node, unless you have had an initial node failure and had to move the controller to another node. When the repository is co-located with the controller it eliminates or greatly reduces the need to compress and transfer data between nodes during the backup process. This is also true with the File Store, especially if your organization uses extracts heavily.

- Use the `--skip-compression` option when backing up Tableau Server. This creates the backup without using compression, and results in a larger backup file but can reduce the amount of time it takes for the backup to complete. For more information, see tsm maintenance backup.

**Perform a Full Backup and Restore of Tableau Server**

You can use the following steps to back up your Tableau Server deployment. Specifically, these steps describe how to recover a clone of a server from a collection of backup data and assets.
Note: The backup process can take a long time to run. Since no other jobs can be run while backup is running, we recommend that you run backup during non-business hours.

Backup data types

There are two types of backup data that Tableau Server can generate. We recommend performing regular backups of each type in case you must restore a server in a recovery scenario:

- Data managed by Tableau Server: consists of the Tableau PostgreSQL database or repository, which contains workbook and user metadata, data extract files, and site configuration data. When you use TSM to create a backup, all of this data is saved in a single file with a .tsbak extension. This data is backed up with the `tsm maintenance backup` command.

  You can only restore from a backup that has the same type of identity store as the running server. For example, a backup from a server using local authentication can be restored to a Tableau Server initialized with local authentication, but a backup from a server using Active Directory authentication cannot be restored to a server initialized with local authentication.

- Configuration and Topology data: includes most of the server configuration information required to fully recover a server. SMTP, alerting, some authentication assets, are all examples of configuration data that are exportable for backup. Topology data defines how your Tableau Server processes are configured in both single-server and multiple node deployments. Configuration and topology data is backed up with the `tsm settings export` command.

Note: You can change the file path used by the `tsm maintenance backup` command from the default value. For more information, see `tsm File Paths`. 
Backup assets that require a manual process

Some configuration data is not included in the `tsm settings export` command and must therefore be documented and restored manually. The following configuration data is excluded from the `tsm settings export` operation. Your backup maintenance process should include documenting the following Tableau Server configuration data:

- **System user accounts.** Tableau Server setup creates an unprivileged user account, `tableau`. This account is used to access Tableau Server resources. This account can be changed during setup. If you have not changed this account, then you do not need to document it.

- **TSM group membership.** There are two groups created by Tableau Server: `tableau` and `tsmadmin`. If you configured alternative groups when you installed Tableau Server, then you’ll need to document the group names.

  In all cases you should document the user accounts that are in these groups. To view membership in a group, run the following command `grep <group_name> /etc/group`.

- **Coordination Service deployment configuration.** If you are running a multinode cluster, document which nodes are running the Coordination Services process. To view process configuration on your nodes, run `tsm topology list-nodes -v`.

- **Customization settings.** If your organization uses custom header or sign-in logos for Tableau Server web pages, you should include a copy of those assets with your backup portfolio. See `tsm customize`.

- **Most authentication assets.** Most certificate files, key files, keytab files or other authentication-related assets are not backed up by TSM. The are three exceptions:

  - The public certificate and private key for the internal PostgreSQL database (if enabled) are backed up.
  - The certificate and key for external SSL are backed up and included in the configuration data.
The custom certificate installed by tsm security custom-cert add (if added) is backed up.

However, all other authentication-related assets are not backed up. For example, if you have enabled access to the PostgreSQL database with the tsm data-access repository-access enable command, be sure to document the name/password pairs for each account you've configured. These credentials are not backed up. The certificate and key for mutual SSL are not included in the back up.

- LDAP assets. Keytab files, configuration files, and or other LDAP-related assets are not backed up by TSM.

Internal server secrets and repository passwords are crypto-related configurations that are not exported. However, you do not need to document configuration values. New secrets will be created as part of the restoration process when you initialize the new instance.

Backing up Tableau Server for recovery

Tableau Server includes commands that you run to generate backup data for Tableau Server.

**Note:** When backing up Tableau Server on Linux, the unprivileged user must have write access to the network share where the backup files are written. Otherwise, backup will fail.

To back up server topology and configuration data, use the tsm settings command.

1. Topology and configuration data are included when you run the `tsm settings export` command. The data is exported as a json file. Specify the name and location of the json file by running the following command:

```bash
    tsm settings export -f <filename>.json
```
Note: Because the backup contains secrets, we recommend that you encrypt the backup and store it in a secure place. For more information about Tableau Server secrets, see Manage Server Secrets.

2. Back up repository data. Repository data is backed up with the `tsm maintenance backup` command. Specify the name and location of the backup file by running the following command:

`tsm maintenance backup -f <filename>.tsbak -d`

By default, the backup file is written here:

```
/var/opt/tableau/tableau_server-
data/tabsvc/files/backups/<filename>.tsbak
```

For more information about where backup files are written, see `tsm File Paths`. For more information about backing up the repository data, see Back up Tableau Server data for more information.

Restoring core Tableau Server functionality

The procedure below uses the assets from the previous two sections to rebuild a Tableau Server in a recovery scenario.

Note: If you need to restore only the repository on an otherwise functional Tableau Server, see Restore from a backup. If you are running a distributed deployment, and your initial node has failed, see Recover from an Initial Node Failure.

Topology and configuration backup data must be from Tableau Server on Linux. You cannot restore configuration data from a backup file that was generated on Tableau Server on Windows. To restore a backup made from Tableau Server on Windows to Tableau Server on Linux, see Migrate Tableau Server from Windows to Linux.
You must have the following assets ready:

- **Topology and configuration data:** This is the json file that is generated by the `tsm settings export` command.

- **Repository backup file:** This is the file with a `.tsbak` extension that is generated by the `tsm maintenance backup` command.

You can only restore from a backup that has the same type of identity store as the running server. For example, a backup from a server using local authentication can be restored to a Tableau Server initialized with local authentication, but a backup from a server using Active Directory authentication cannot be restored to a server initialized with local authentication.

- **Backup assets:** These assets include the list of documented configurations as noted in the previous section.

**To restore a standalone Tableau Server**

1. On the computer where you want to restore Tableau Server, Install and Initialize TSM. If your organization used non-default system user accounts, as described in an early section of this topic, then you must specify the users during this step.

2. Activate and Register Tableau Server.

3. (Optional). Configure Local Firewall.

4. Import topology and configuration data. Copy the topology and configuration json backup file to the computer. Import the json file by running the following command:

   ```bash
tsm settings import -f <filename>.json
   ```

5. (Optional). Apply pending changes. At a command prompt, run:
tsm pending-changes apply

6. Restart Tableau Server. At a command prompt, run:

    tsm restart

7. Restore repository data. See Restore from a backup.

8. (Optional). Repopulate TSM group membership. Add users to groups with this command:

    sudo usermod -G <group_name> -a <username>

To restore a Tableau Server cluster

1. On the initial node, Install and Initialize TSM. If your organization used non-default system user accounts, as described in an early section of this topic, then you must specify the users during this step.

2. On the initial node, Activate and Register Tableau Server.

3. (Optional). On the initial node, Configure Local Firewall.

4. On the initial node verify LDAP (optional), and initialize Tableau Server. See Configure Initial Node Settings.

5. On the initial node, run tsm topology nodes get-bootstrap-file --file <path\file>.json.

6. Copy the bootstrap.json file to all additional nodes in the cluster.

7. On each additional node in the cluster:

   a. Install the Tableau Server package.

   b. Navigate to the scripts directory.
c. Initiate communication between initial node and the additional node:

```
sudo ./initialize-tsm -b <path-to-bootstrap>.json -u <admin-user-on-first-node> --accepteula
``` 

8. On the initial node, run `tsm topology list-nodes -v` and ensure that the node names have not changed from exported topology settings. If they have, topology settings should be manually updated with new names, or the processes should be manually configured.

9. On the initial node, Deploy a Coordination Service Ensemble. The ensemble configuration must match your previous configuration.

10. On the initial node, import topology and configuration data. Copy the topology and configuration json backup file to the computer. Import the json file by running the following command:

```
tsm settings import -f <filename>.json
```

11. On the initial node, apply pending changes. At a command prompt, run:

```
tsm pending-changes apply
```

12. On the initial node, restart Tableau Server. At a command prompt, run:

```
tsm restart
```

13. On the initial node, restore repository data. See Restore from a backup.

14. On the initial node, repopulate TSM group membership. Add users to groups with this command:

```
sudo usermod -G <group_name> -a <username>
```

Restore other functionality

If the previous server was configured with the following features, then you will need to re-enable and reconfigure them on the restored server:
• Authentication solutions: OpenID, external SSL, and trusted authentication. See Authentication.

• Site customizations: See tsm customize.

• Enable access to PostgreSQL repository: See tsm data-access repository-access enable.

Reencrypt Extracts After Restore

Optionally, if you are using the extract encryption at rest feature, after the backup is restored, you can reencrypt the extracts using different encryption keys. See Extract Encryption at Rest.

Run `tabcmd reencryptextracts <site-name>` to reencrypt extracts on a given site. For more information, see reencryptextracts. Run this command on every site where you are storing encrypted extracts. Depending on the number of encrypted extracts on the site, this operation could consume significant server processing load. Consider running this operation outside of business hours.

Back up Tableau Server data

Regularly backing up Tableau Server is an important step in proper administration and maintenance of your server. You can use the `tsm maintenance restore` command to restore Tableau Server backups created using `tabadmin backup` and `tsm maintenance backup`. Database backups made in other ways, and virtual machine snapshots are not valid sources for restoring Tableau Server, so it is critical that you have an up-to-date backup.

You can only restore from a backup that has the same type of identity store as the running server. For example, a backup from a server using local authentication can be restored to a Tableau Server initialized with local authentication, but a backup from a server using Active Directory authentication cannot be restored to a server initialized with local authentication.
Tableau Server on Linux Administrator Guide


Tableau Server data includes data extract files, as well as the Tableau PostgreSQL database, which contains workbook and user metadata, and configuration data. When you use TSM to create a backup, all of this data is saved in a single file with a .tsbak extension. If you are running a distributed installation of Tableau Server data from all the nodes is backed up.

The frequency of your backups depends on your environment, including how much use your server gets and how much and frequently the content and users change. Any changes or updates that happen after your backup will be lost if there is a system failure and you need to restore Tableau Server. The more activity there is, the more often you need to back the server up.

In addition to regular backups, you should always create a backup before upgrading to a new version of Tableau Server.

To help protect against data loss, after you create the backup, you should store the .tsbak file on a computer that is not a part of your Tableau Server installation.

Disk Space Usage for Backup

Backup disk space requirements

You need sufficient free disk space to restore from backup, that is in all cases 15% of the disk size. In some cases it could be even greater. During backups, the background tasks for cleaning up old extracts are temporarily paused. This means that, for the duration of the backup, extract refreshes will leave extra files in place, adding to disk space usage. If your backup takes a long time, or if your organization uses many extracts that are regularly updated, this can result in a significant amount of temporary disk space usage. These old files will be removed after the backup is complete.
On nodes with the repository, you need additional free space equal to at least twice the size of the repository data. To obtain an estimate of the repository data, check the size of `<data directory>/pgsql/data/base directory`.

On nodes with the file store, you need additional free space equal to at least the size of the file store data. To obtain an estimate of file store data (extracts, flows, etc.), check the size of `<data directory>/dataengine directory`.

On nodes with the Administration Controller or TSM Controller (usually the initial node unless you have had to recover from an initial node failure), you need additional free space equal to at least the size of the repository data plus the size of the file store data.

If all of the above are on the same node, you can use the sum of the sizes to determine the amount of disk space required.

Create a backup using the TSM command line interface (CLI)

Use the `tsm maintenance backup` command to create a backup of the data managed by Tableau Server. This data includes data extract files and the Tableau PostgreSQL database, which contains workbook and user metadata.

**Note:** When backing up Tableau Server on Linux, the unprivileged user must have write access to the network share where the backup files are written. Otherwise, backup will fail.

To back up server configuration data, use the `tsm settings` command. When you use the `tsm maintenance backup` command, the current date is appended to the backup file:

```
tsm maintenance backup -f <backup_file> -d
```

For more information, see `tsm maintenance backup`. 
Create a pre-upgrade backup

You should always create a backup before upgrading Tableau Server. You can create a backup while Tableau Server is running and minimize the amount of time the server is unavailable during upgrade. The process for creating a pre-upgrade backup is the same as for creating regular backups, with one additional consideration for distributed installations.

**Note:** Uninstall Tableau Server from any nodes that you are not including in your new installation to avoid conflicts between the older nodes and the new installation.

Script the backup process

If you back up often, you might want to create a script that performs the backup and related tasks for you. These tasks include:

- Clean up files and folders before running the backup.
- Running the backup itself.
- Copying the backup file to a separate computer for safekeeping.

This section discusses `tsm` commands you can use together to perform a backup and related tasks.

For additional information about scripting TSM commands, see the Tableau Community Forums.

Remove log files and clear temporary folders

You can clean old Tableau Server log file and temporary files to reduce the time it takes to create a backup, and to ensure the backup file is as small as possible.

To clean log files older than a few days, run the following command:

```
  tsm maintenance cleanup
```
Run the backup

**Note:** When backing up Tableau Server on Linux, the unprivileged user must have write access to the network share where the backup files are written. Otherwise, backup will fail.

To create the backup, use the `tsm maintenance backup` command:

```
  tsm maintenance backup --file <backup_file> --append-date
```

Note the following about the command:

- **Add** `--append-date` to the command to include the date in the backup file name.
- **By default** the backup file is created in `/var/opt/tableau/tableau_server-data/tabsvc/files/backups/`. For more information, see `tsm maintenance backup`.

Copy the backup file to another computer

As a best practice, after the backup is created, copy the backup file to another location that is separate from Tableau Server.

**Restore from a backup**

Use the `tsm maintenance restore` command to restore your Tableau Server data. You might do this if you had a system failure and need to restore your data, if you need to switch back to a previous version of Tableau Server (for example, if there is a problem with an upgrade), or if you are moving Tableau Server to new hardware. You can use the `tsm maintenance restore` command to restore Tableau Server backups created using `tabadmin backup` and `tsm maintenance backup`. If you are restoring a backup created using `tabadmin backup`, **and you used a custom asset key**, you must save a copy of your `asset_keys.yml` file so you can include the file when doing your restore. For more information, see Save your asset keys file before uninstalling Tableau Server for Windows 2018.1.x or earlier.
Database backups made in other ways, and virtual machine snapshots are not valid sources for restoring Tableau Server.

When you use `tsm maintenance restore` to restore your Tableau data, data extract files and the contents of the PostgreSQL database are overwritten with the content in the backup file (.tsbak). If you are running a distributed installation of Tableau Server, perform the restore on the initial node or where ever the TSM Controller is running.

You can only restore from a backup that has the same type of identity store as the running server. For example, a backup from a server using local authentication can be restored to a Tableau Server initialized with local authentication, but a backup from a server using Active Directory authentication cannot be restored to a server initialized with local authentication.

**Restore Tableau Server from a backup file**

**Note:** This operation includes steps that you may need to perform using the TSM command line.

1. **(Optional)** Copy the `.tsbak` file to the default file location.

   **Note:** If you are restoring a backup that was copied into the backups folder, ensure that the Run As service account, which can be found in TSM Web UI under Security, has at least Read access to the backup file. Otherwise, the restore process may not be able to unzip the backup file and the restore will fail.

2. Stop the server. At a command prompt, type:

   ```
   tsm stop
   ```

3. Restore from a backup file. At a command prompt, type:

   ```
   tsm maintenance restore --file <file_name>
   ```
In the above line, replace `<file_name>` with the name of the backup file you want to restore from.

**Note:** If you encounter errors when trying to restore from backup, see Troubleshoot Tableau Server on Linux.

4. Restart the server:

```bash
tsm start
```

When you restore a `.tsbak` file, Tableau Server automatically creates a copy of its current data folder, names it `tabsvc.bak-*`, and places it in `/var/-opt/tableau/tableau_server/data`. This folder is an emergency backup of Tableau Server that Tableau Support may be able to use in case something goes wrong during backup restoration.

When the restore is complete and you have verified that Tableau Server is running correctly with all the expected data, it's safe to remove any `tabsvc.bak-*` folders from `/var/-opt/tableau/tableau_server/data` to free additional disk space. In Tableau Server clusters, `tabsvc.bak-*` folders are created on each machine running Tableau Server.

**Important:** Only remove the `tabsvc.bak-*` folders. Do not remove the `tabsvc` folder, which is also located under `/var/opt/tableau/tableau_server/data`. It contains necessary Tableau Server data.

**Server Maintenance**

As an administrator, you will want to check the status of the server, analyze and monitor the activity on the server, manage scheduled tasks, or perform certain maintenance activities such as clearing saved data connection passwords. In addition, there are several settings that you may want to specify to customize the user experience for people using the server.
You can do some of these tasks from the General page of the Status page and others from the Settings page.

View Server Process Status

You can view server process status can be by running a TSM CLI command or by accessing TSM Web UI or Admin pages on Tableau Server.

Viewing process status with TSM CLI

Run the following command:

```
tsm status -v
```

This command outputs all of the processes that are configured on the instance and their corresponding status.

Viewing process status in web UI

There are two locations in Tableau Server or Tableau Services Manager (TSM) where administrators can view the state of Tableau processes. You may be able to access one or both of these locations, depending on how your account and server are set up. Most of the process status information that displays is duplicated on both Status pages. This section explains each page, and identifies what is unique for each one.

**Note** The status pages displayed in both web UI locations show a subset of the total processes configured on a given node. To view all processes, you must run the TSM CLI command, `tsm status -v`.

- The Tableau Services Manager (TSM) status page is accessible in TSM and can be viewed by TSM administrators. You must be able to log into TSM to see this page. For information about signing into TSM, see Sign in to Tableau Services Manager Web UI.

- The Tableau Server status page appears in the Tableau Server web UI and is
accessible by Tableau Server administrators. This page includes Tableau Server processes, along with links to troubleshooting documentation if a process is not running as expected. If you hover your mouse pointer over the status indicator for a process, a tooltip shows the node name and the port the process is running on. The Tableau Server status page does not show TSM processes. For information about signing into Tableau Server as an administrator, see Sign in to the Tableau Server Admin Area.

Tableau Services Manager (TSM) Status page

The TSM Status page shows you the state of server processes, including TSM Controller and License Server. These two processes do not display on the Tableau Server Status page.
Possible status indicators are listed at the bottom of the table:

- **Active**
- **Busy**
- **Degraded**
- **Error**
- **Stopped**
Tableau Server Status page

Tableau Server Status page

Tableau Server Status

Table: Process Status

<table>
<thead>
<tr>
<th>Process</th>
<th>OPENVM-QKRAFP59</th>
<th>OPENVM-28F71269</th>
<th>OPENVM-QKRAFPEL</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cluster Controller</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>Gateway</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>Application Server</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>ViSQL Server</td>
<td>✔ ✔ ✔</td>
<td>✔ ✔</td>
<td>✔ ✔</td>
</tr>
<tr>
<td>Cache Server</td>
<td>✔ ✔ ✔</td>
<td>✔ ✔</td>
<td>✔ ✔</td>
</tr>
<tr>
<td>Search &amp; Browse</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>Backgrounder</td>
<td>✔ ✔</td>
<td>✔ ✔</td>
<td>✔ ✔</td>
</tr>
<tr>
<td>Data Server</td>
<td>✔ ✔</td>
<td>✔ ✔</td>
<td>✔ ✔</td>
</tr>
<tr>
<td>Data Engine</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>File Store</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>Repository</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
</tbody>
</table>

Possible status indicators are listed at the bottom of the table:

- **Active**—The process is functioning as intended. See File Store in Troubleshoot Server Processes for details on possible active states.
- **Busy**—The process is completing some task. See File Store and Repository in Troubleshoot Server Processes for more information.
- **Passive**—The repository is in passive mode
Tableau Server on Linux Administrator Guide

- **Unlicensed**—The process is unlicensed.
- **Down**—The process is down. The implications of this differ depending on the process.
- **Status unavailable**—Tableau Server is unable to determine the status of the process.

If there is additional information, a message displays below the status icon and links to appropriate documentation:

![Ready for removal](image)

For more information about troubleshooting process status, see Troubleshoot Server Processes.

Access Status Remotely

**Note:** The information in this article refers to the Tableau Server status page. For information about the Tableau Server status page and the TSM status page, see View Server Process Status.

You must be a Tableau Server administrator to see the Server Status page, but you can grant remote access to other computers to allow access to a machine-readable (XML) version of the Status table by non-admin users and by computers other than the initial Tableau Server node. One reason you might do this is as part of a remote monitoring process.

To grant remote access to Tableau Server status:

1. Open a command prompt as an administrator and type the following:

   ```
tsm configuration set -k wgserver.systeminfo.allow_referrer_ips -v <ip address>
   ```

   In the above command, `<ip address>` is the IPv4 address of the computer for which you want to enable remote access to the Tableau Server status XML.
For example:

tsm configuration set -k wgserver.systeminfo.allow_referer_ips -v 10.32.139.31

If you are enabling remote access for more than one computer, use commas to separate each IP address.

tsm configuration set -k wgserver.systeminfo.allow_referer_ips -v 10.32.139.31,10.32.139.35

2. Commit the configuration change:

tsm pending-changes apply

The `pending-changes apply` command displays a prompt to let you know this will restart Tableau Server if the server is running. The prompt displays even if the server is stopped, but in that case there is no restart. You can suppress the prompt using the `--ignore-prompt` option, but this does not change the restart behavior. For more information, see `tsm pending-changes apply`.

Now, users of computers with the IP addresses that have been added can view Tableau process status by entering the URL `http://<server>/admin/systeminfo.xml` in a browser or from a command line (for example, `curl http://jsmith/admin/systeminfo.xml`).

If Tableau Server has been configured to work with a load balancer or proxy server, use the hostname or IP address of the initial Tableau Server node to access the XML version of the status page.

For details on the XML that is returned, see Get Process Status as XML.

Get Process Status as XML

To get a machine-readable version of the server process status, that is, a version of the status formatted in XML, use the following URL:

http://my_tableau_server/admin/systeminfo.xml
You must be signed in to Tableau Server to view the machine-readable process status, or have enabled remote access.

The server returns a status report similar to the following:

```xml
<systeminfo xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
  <machines>
    <machine name="my_tableau_server">
      <repository worker="my_tableau_server:8060" status="Active" preferred="false"/>
      <dataengine worker="my_tableau_server:27042" status="Active"/>
      <applicationserver worker="my_tableau_server:8600" status="Active"/>
      <apiserver worker="my_tableau_server:8000" status="Active"/>
      <vizqlserver worker="my_tableau_server:9100" status="Active"/>
      <dataserver worker="my_tableau_server:9700" status="Active"/>
      <backgrounder worker="my_tableau_server:8250" status="Active"/>
      <gateway worker="my_tableau_server:80" status="Active"/>
      <searchandbrowse worker="my_tableau_server:11000" status="Active"/>
      <cacheserver worker="my_tableau_server:6379" status="Active"/>
      <filestore worker="my_tableau_server:9345" status="Active" pendingTransfers="0" failedTransfers="0" syncTimestamp="2015-02-27T20:30:48.564Z"/>
      <clustercontroller worker="my_tableau_server:12012" status="Active"/>
      <coordination worker="my_tableau_server:12000" status="Active"/>
    </machine>
  </machines>
</systeminfo>
```
Status values in the XML

- **<process> worker** - The name of the node running the process and the port the process is using.

- **status** - The status of the process on the node. Possible values are: Active, Passive, Unlicensed, Busy, Down, ReadOnly, ActiveSyncing, StatusNotAvailable, StatusNotAvailableSyncing, NotAvailable, DecommissionedReadOnly, DecomissioningReadOnly, and DecomissionFailedReadOnly

- **pendingTransfers** - A count of the workbook or data source extracts the node needs to get to be fully synced. These represent items that were published to this file store node, and items that were published to other file store nodes and need to be copied to this node.

- **failedTransfers** - A count of the workbooks or data sources that did not transfer successfully to this file store node during the last automated job. The automated job normally runs about every 15 to 30 minutes, but may take longer when transferring a large number of extracts or large extracts.

  Failed transfers do not necessarily indicate a problem with Tableau Server. The recurring automated job will normally transfer files that failed during the previous sync. Reasons for failed file transfers are listed in the logs.

- **syncTimestamp** - The time in UTC of the last automated job that ran and synchronized files.
Troubleshoot Server Processes

**Note:** The information in this article refers to the Tableau Server status page. For information about the Tableau Server status page and the TSM status page, see View Server Process Status.

When Tableau Server is functioning properly, processes will show as Active, Busy or Passive (Repository). If there is additional information, a message appears below the status icon:

![Ready for removal](image)

Possible status indicators are:

![Status icons](image)

The following sections provide troubleshooting recommendations for status messages that you may see.

**Cluster Controller**

This message will only display if you have more than two nodes.

**Status: Down; Message: "Node degraded"**

One or more of the following are true:

- Repository on the node is stopped.
- Node cannot respond to failover elsewhere on the cluster.
- If Tableau Server is configured for high availability and this is the active repository, failover to the second repository occurs.
- No status available for repository or file store on this node.
No action is necessary unless the cluster controller is regularly down or is down for an extended period of time.

If that occurs, take the following actions, in order, until the problem is resolved:

1. Check disk space. If disk space is limited, save the log files (use `tsm maintenance ziplogs`) in case you need them for Support, then remove unnecessary files.
2. Restart Tableau Server.
3. If Cluster Controller continues to show as down, save the log files (`tsm maintenance ziplogs`) and contact Support.

File Store

File Store status only reflects the state of the file store when the page was loaded.

An active status (✓) with no message indicates that no extracts were being synchronized when the page was loaded. It is possible that the recurring "catch-all" job is running and synchronizing extracts.

적이지 않은 상태 (✓)와 함께 메시지가 없는 경우에는 파일 스트어의 상태가 계산된 이 경우에만 유효합니다. 이는 재생성하는 작업이 실행되고 있는지 알 수 있습니다.

Status: Busy; Message: "Synchronizing"

"Synchronizing," usually indicates that extracts were being synchronized across file store nodes when the page was loaded.

However, the "synchronizing" message is also returned following installation (both single-node and multi-node). After Tableau initializes the status should disappear within 15 or 20 minutes.

Status: Down; "Data Extracts unavailable"

On a single-node installation: "Data Extracts unavailable" indicates that existing extracts may be available but publish/refresh will fail. On multi-node installations, this message indicates that extract synchronization will fail for this node.

No action is necessary unless the file store is regularly down or is down for an extended period of time.
Tableau Server on Linux Administrator Guide

If that occurs, take the following actions, in order, until the problem is resolved:

1. Check disk space. If disk space is limited, save the log files (use tsm maintenance ziplogs) in case you need them for Support, then remove unnecessary files.
2. Restart Tableau Server.
3. If Cluster Controller continues to show as down, save the log files (tsm maintenance ziplogs) and contact Support.

⚠️ Status: Busy; "Decommissioning"

This message indicates that this File Store is in read-only mode and that any unique files on this node are being replicated to other file store nodes.

To remove this node, wait until the status message changes to "Ready for removal".

✅ Status: Active; "Ready for removal"

This message indicates that the File store is in read-only mode.

You can safely stop (tsm stop) the cluster and remove File Store processes, or remove entire node.

✅ Status: Active; "Decommission failed"

This message indicates that the File store is in read-only mode, and that at least one unique file failed to replicate to another file store node.

To resolve a failed decommissioning:

1. Run the tsm topology filestore decommission command again.
2. Check disk space on other file store nodes. Decommissioning will fail if another file store node does not have enough space to store all the extracts.
3. Check the tsm.log file on the initial node and additional nodes for errors.
4. Stop Tableau Server (tsm stop) and then try running the tsm topology filestore decommission command again.
5. Put the file store node back into read/write mode (tsm topology filestore recommission), collect logs, and then contact Support.
6. With Support: copy and merge extracts directory from this file store node to the same directory on another file store node.

Repository

**Status: Busy; Message: "Setting up"**

The "Setting up" message indicates one or more of the following states:

- Passive repository is being synchronized with active repository.
- Repository is not ready to handle failover.
- Repository may have gotten more than two minutes behind active repository and is being setup again (this is faster than waiting for a sync).
- Failover occurred and this former active repository is rejoining the cluster.

Wait until the repository status message changes to "Passive".

If this message does not appear, or if it is taking a long time:

1. Check disk space and free space if possible.
2. Check cluster controller logs for errors.
3. Restart node.

**Status: Busy; Message: "Synchronizing"**

Repository is synchronizing, for example after a failover.

**Status: Down; Message: n/a**

When the Repository shows a status of down and there is no message, then the Repository is in one of the following states:

- If the installation is configured for high availability, failover of the repository occurred.
- Processes are restarting with updated database connection configurations after failover.
- If another active repository is not available, Tableau Server is down.

Take these actions in order until a step resolves the problem:
1. Wait several minutes for cluster controller to attempt to restart.
2. Restart Tableau Server (tsm restart).
3. Check disk space. If disk space is limited, save the log files (use tsm maintenance ziplogs) in case you need them for Support, then remove unnecessary files.
4. Restart Tableau Server.
5. If repository continues to show as down, save the log files (tsm maintenance ziplogs) and contact Support.

✅ **Status: Passive; Message: n/a**

A passive status with no message indicates that the node is working as intended and that it is ready for failover if needed.

**VizQL Server**

⚠️ **Status: Unlicensed; Message: n/a**

For information about unlicensed status for a VizQL Server process, see Handle an Unlicensed VizQL Server Process.

**Clear Saved Data Connection Passwords**

As the administrator, if you enable users to save data source passwords, server users can save data source passwords across multiple visits and browsers so they are no prompted for their credentials each time they connect to a data source.

You can reset the data source passwords for all Tableau Server users. Doing this forces them to sign in to the data sources the next time they visit a view that requires database authentication. Server users can also clear their saved data connection passwords on an individual basis using their User Preferences page.

To clear saved data connection passwords for all server users:

1. In a single-site server, click **Settings > General**. On a multi-site server, select **Manage all sites**, then click **Settings > General**.
2. Under Saved Credentials, click **Clear All Saved Credentials**.

Synchronize All Active Directory Groups on the Server

As a server administrator, you can synchronize all Active Directory groups (that have been configured on Tableau Server) on a regular schedule or on-demand on the **General** tab of the **Settings** page for the server.
**Note:** If you run Tableau Server on Linux, all external directory communication is configured and managed with a LDAP identity store. In the context of user and group synchronization, Tableau Server configured with LDAP identity store is equivalent to Active Directory. Active Directory synchronization features in Tableau Server function seamlessly with properly configured LDAP directory solutions.

Before you begin

Before synchronizing groups as described in this topic, you must first import the Active Directory group into Tableau Server. See Create Groups via Active Directory.

Synchronize Active Directory groups on a schedule

1. **Single-site:** Click **Settings** > **General**.

   **Multisite:** In the site menu, click **Manage All Sites** and then click **Settings** > **General**.

2. Scroll down the page to **Active Directory Synchronization**, and then select **Synchronize Active Directory groups on a regular schedule**.
3. Select the frequency and time of synchronization.

4. Click **Save**.

Synchronize all Active Directory groups on demand

At any time, you can synchronize Active Directory groups with Tableau Server to ensure that new users and changes in Active Directory are reflected in all Active Directory groups on Tableau Server.

1. **Single-site**: Click **Settings**> **General**.

    **Multisite**: In the site menu, click **Manage All Sites**, and then click **Settings**> **General**.
2. Under **Active Directory Synchronization**, click **Synchronize All Groups**.

View synchronization activity

You can view the results of synchronization jobs in the **Background Tasks for Non Extracts** administrative view. **Queue Active Directory Groups Sync** is the task that queues and indicates the number of **Sync Active Directory Group** tasks to be run.

1. **Single-site**: Click **Status**.

   **Multisite**: In the site menu, click **Manage All Sites** and then click **Status**.

2. Click the **Background Tasks for Non Extracts** link.

3. Set the **Task** filter to include **Queue Active Directory Groups Sync** and **Sync Active Directory Group**.

You can quickly navigate to this administrative view by clicking the **View synchronization activity** link in the **Settings** page for the server.

Set the minimum site role for users in an Active Directory group

In the **Groups - Details** page, you can set the minimum site role for group users to be applied during Active Directory synchronization.
This setting does not run synchronization; instead, it sets the minimum site role to applied to the group every time synchronization runs. The result is that when you synchronize Active Directory groups, new users are added to the site with the minimum site role. If a user already exists, the minimum site role is applied if it gives the user more access in a site. If you don't set a minimum site role, new users are added as Unlicensed by default.

**Note:** A user’s site role can be promoted but never demoted based on the minimum site role setting. If a user already has the ability to publish, that ability will always be maintained. For more information on minimum site role, see Site roles and Active Directory import and synchronization.

1. In a site, click **Groups**.
2. On the Groups page, select a group. Click **Actions > Minimum Site Role**.
3. Select the minimum site role, and then click **Change Site Role**.
What happens when users are removed in the source Active Directory?

Users cannot be automatically removed from the Tableau Server through an Active Directory sync operation. Users that are disabled, deleted, or removed from groups in Active Directory remain on Tableau Server so that administrators can audit and reassign the user’s content before removing the user’s account completely. For more information, see Sync behavior when removing users from Active Directory.

Improving group synchronization performance

Active Directory synchronization is performed by the backgrounder process. The Backgrounder process is the same process that is used for managing and creating extracts, and is also used to generate subscription content. In large organizations with dynamic group membership and heavy extract usage, the Active Directory group synchronization process may be disruptive. We recommend running group synchronization during non-business hours.

By default, the Backgrounder process performs synchronization in a serial operation. This means that each group is synchronized, one after the other, in a single Backgrounder process. If you are running multiple instances of Backgrounder processes either on a single Tableau Server or across a distributed deployment, consider enabling parallel processing for Active Directory synchronization. When parallel Backgrounder processing is enabled, the group synchronization is distributed across multiple Backgrounder processes for better performance.

To enable parallel backgrounder processing for group synchronization, open TSM CLI and enter the following commands:

```bash
  tsm configuration set -k backgrounder.enable_parallel_adsync -v true
  tsm pending-changes apply
```

Set the Default Start Page for All Users

In a new deployment, when users sign in to the Tableau Server web authoring environment, they are taken to the Home screen, which displays a role-based welcome banner. Home
also displays recent views, favorites, and the site's most-viewed content. As the server administrator, you can change users’ default landing page at the server and site level. For example, you can show all workbooks, and when the user signs in, they see the workbooks they have access to.

To set the default start page for all users

1. Display the page or filtered view you want to be the default page users see when they sign in to the site.

2. Select your profile icon in the upper right area of the page.

3. To set the start page for:
   - All users on a site: click **Set as Start Page** and **For Users on this Site**.
   - All users on the server: click **Set as Start Page** and **For All Server Users**.

User-set start pages and hierarchy

Users can set their own start pages from their profile icon, and can reset their start pages in their account settings (for information, search for “Access Your Profile and Account Settings” in the online Tableau Server Help for your operating system).

If a user sets their own start page, it will override any start page set by a server administrator. The next time that user signs in, they will land on the start page they've set. If a server administrator sets start pages for both a server and a site, users will default to the
Disable Automatic Client Authentication

After Tableau Desktop and Tableau Mobile clients first successfully sign in to Tableau Server, they are automatically authenticated in the future. Client sessions are managed by OAuth access and refresh tokens.

By default, all client tokens reset after a year. If a client token has not been used in 14 days, then it will expire. You can change these values by setting the `refresh_token.-absolute_expiration_in_seconds` and `refresh_token.idle_expiration_in_seconds` options. See tsm configuration set Options.

Tableau Server administrators can also disable automatic authentication. To immediately disconnect all clients from Tableau Server and require users to sign in every time they connect:

1. In the site menu, click Manage All Sites, and then click Settings > General.

2. Under Connected Clients, clear the option Allow clients to automatically connect to Tableau Server.

3. Click Save.

Remove Unneeded Files

As a best practice, you should monitor space usage on your server. If you need to make more space available, you can use the tsm maintenance cleanup command to remove Tableau Server log files, temporary files, and rows from the http_requests table of the Tableau Server PostgreSQL database.

You should run the tsm maintenance cleanup command regularly.
Server Settings (General and Customization)

The following settings are available on the General and Customization pages in Server - Settings.

General

<table>
<thead>
<tr>
<th>Setting</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Language and Locale</td>
<td>Controls the language used for the server user interface and the locale used for views. Individual users can override this setting on their Account Settings page. Also, web browser settings are evaluated first to determine which language and locale should be used. For more information, see Language and Locale for Tableau Server.</td>
</tr>
<tr>
<td>Start Page</td>
<td>Links to the server’s current default start page for all users. For more information on how to change the default start page, see Set the Default Start Page for All Users. Individual users will be able to override this setting (search for &quot;Access Your Profile and Account Settings&quot; in the Tableau Server Help for details).</td>
</tr>
<tr>
<td>Connected Clients - Let clients automatically connect to Tableau Server</td>
<td>Controls whether mobile users must sign in and provide their credentials every time they connect to Tableau Server, or if users can connect with their devices to Tableau Server without providing credentials after they authenticate their device successfully the first time. For more information, see Disable Automatic Client Authentication.</td>
</tr>
<tr>
<td>Active Directory Synchronization - Synchronize Active Directory</td>
<td>Controls the synchronization of all Active Directory groups in Tableau Server based on a</td>
</tr>
</tbody>
</table>
### groups on a regular schedule

Schedule that you specify after you select the option **Synchronize Active Directory groups on a regular schedule**. For more information, see Synchronize All Active Directory Groups on the Server.

### Guest Access - Enable guest access

Allows users to view and interact with embedded views without having to sign in to a Tableau Server account. Permission can be assigned to the Guest User account to control the interactivity allowed for each view. This option is only available if you have a core-based server license.

### Embedded Credentials in Content - Let publishers embed credentials in a data source, flow, or workbook

Allows publishers to attach passwords to published workbooks or flows that will automatically authenticate web users to connect to data sources. The passwords are attached to workbooks or flow inputs and are only accessible on the server. For example, when a workbook is opened in Tableau Desktop, users still need to enter a user name and password to connect to the data source. When this setting is turned off, all existing embedded passwords are saved but are not used for authentication. If you turn the setting back on, users don’t have to re-embed the passwords.

### Embedded Credentials in Content - Let publishers schedule flow runs and extract refreshes

Allows publishers to assign workbooks or flows to schedules. This option is only available if **Let publishers embed credentials in a data source, flow, or workbook** is enabled. When this setting is enabled, Tableau Desktop users will see scheduling options in the Publish dialog box.

### Saved Credentials for Data

Allows users to choose "Remember my pass-
| **Sources - Let users save passwords for data sources** | word" and save data source passwords across multiple visits, browsers, and devices. (By default, users can choose to "Remember my password until I sign out," which lets them save their password only for a single browser session.)

As an administrator, you can clear all saved passwords at any time. In addition, users can clear their own saved passwords. |
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Saved Credentials for Data Sources - Let users save OAuth access tokens for data sources</strong></td>
<td>Allows users to store access tokens with their user preferences. Access tokens are provided by cloud data sources that support OAuth connections, and they are used instead of user names and passwords to grant access to the data.</td>
</tr>
</tbody>
</table>
| **Recommendations Training Schedule** | Recommendations take two forms: recommendations for data sources and tables (for Tableau Desktop) and recommendations for views (for Tableau Server). Recommendations are based on the popularity of content and on content used by other users determined to be similar to the current user.

The training schedule controls how often the server checks for new content and new usage information to keep the recommendations up to date. New content includes new and updated data sources and workbooks. New usage information includes information such as "Laura Rodriguez used the Food Catering data source" and "Henry Wilson accessed the Monthly Sales view." |
<table>
<thead>
<tr>
<th>If you notice an impact on server performance, schedule this process to occur at a time when the server load is low. To track performance impact, look for the &quot;Recommendations Trainer&quot; or &quot;View Recommendations Trainer&quot; tasks in the Background Tasks for Non Extracts view. If you want to disable Recommendations, change the <code>tsm configuration set</code> option &quot;recommendations.enabled&quot; to false. If you want to disable only recommendations for views, change the option &quot;recommendations.vizrecs.enabled&quot; to false.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Sign In</strong></td>
</tr>
<tr>
<td><strong>Reset to Default Settings</strong></td>
</tr>
</tbody>
</table>
Customization

<table>
<thead>
<tr>
<th>Setting</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Welcome Banner</td>
<td>Add a custom message to the welcome banner on the home page for all server users to see. The custom message can contain up to 240 characters of text and hyperlinks as well as one paragraph break. Administrators can also disable the default Tableau welcome banner for the server. For more information, see Customize Your Server.</td>
</tr>
</tbody>
</table>

Stop or Restart the Tableau Server Computer

As a best practice, you should always stop Tableau Server before you stop or restart the computer it is running on. This is true whether you are running Tableau on virtual machines (VMs), or on dedicated hardware. You should never turn off a computer without first stopping Tableau Server. Shutting down the computer while Tableau is running can cause problems restarting Tableau Server, and may result in unexpected results.

To be safe, follow these steps, whether you have a scripted process to shut down your systems, or manually shut down your computers:

1. Stop Tableau Server.

   You can do this either from the command line, using the `tsm stop` command, or from the TSM Web UI, by clicking **Tableau Server is running**, and selecting **Stop Tableau Server**.
Note: Some TSM processes will continue to run, even after you stop Tableau Server. This is normal, and you can go ahead and stop your computer. The running services are designed to shut themselves off when the computer is stopped.

2. Once Tableau is stopped, stop your computer.

3. When you are ready, restart your computer. This might be after you have completed planned maintenance, or after leaving the computer off for some extended period of down time.


You can do this either from the command line, using the `tsm start` command, or from the TSM Web UI, by clicking **Tableau Server is stopped**, and selecting **Start Tableau Server**.

TSM will start automatically when the computer starts, so you can run tsm commands even though Tableau Server is stopped.

**tsm Command Line Reference**

The topics in this section include reference content for Tableau Services Manager (TSM) command line interface (CLI) to support Tableau Server.

TSM is used to manage installation and configuration of Tableau Server. To learn more about TSM, see Tableau Services Manager Overview.

You can automate the installation and configuration tasks supported by the TSM CLI using the TSM API. To learn more about the prerelease (Alpha) TSM API, see **Tableau Services Manager API**.

Looking for tsm commands for Tableau Server on Windows? See **tsm Commands**.
Using the tsm CLI

You can run tsm commands on the initial node (the node where TSM is installed), or on any additional node in the cluster.

To run tsm commands, you need to open a command prompt.

1. Open a command prompt with an account that is a member of the tsmadmin group on a node in the cluster.

2. Run the command you want. If you are running the command from a node other than the initial node, include the -s option to specify the URL of the initial node by name (not IP address), and include the TSM port, 8850.

To see the version of TSM and Tableau Server from the initial node:

```plaintext
tsm version
```

To see the version of TSM and Tableau Server from an additional node:

```plaintext
tsm version -s https://<initial_node_name>:8850
```

For example:

```plaintext
tsm version -s https://myTableauHost:8850
```

Authenticating with tsm CLI

Beginning in the 2019.2 release of Tableau Server, running tsm commands will not require you to enter a password if the following are true:

- The account you are running commands with is a member of the TSM-authorized group, by default, the tsmadmin group. The Tableau unprivileged user (by default, the tableau user) and root account may also run TSM commands.
- You are running commands locally on the Tableau Server that is running the Tableau Server Administration Controller service. By default, the Tableau Server Administration Controller service is installed and configured on the initial node in a distributed deployment.
Logging into tsm CLI locally

If you are running tsm commands on the local computer with user account that is a member of a TSM-authorized group, then you will not need to specify a password. In this case, just run the command, for example:

```bash
 tsm version
```

Logging into tsm CLI remotely

If you are running TSM commands from a node in a cluster where the Tableau Server Administration Controller service is not running, then you must authenticate a session with the Tableau Server Administration Controller service on the remote computer before you can run commands. For example, run the following command:

```bash
 tsm login -s <server_name> -u <account_name>
```

Where `<server_name>` is the name of the node where the Tableau Server Administration Controller service is running and `<account_name>` is an account that is a member of a TSM-authorized group.

After running this command, you will be prompted for a password. After the account has been authenticated, you can run TSM commands.

Viewing and adding accounts to the TSM-authorized group

The TSM-authorized group is created during server installation. By default, the TSM-authorized group that is named `tsmadmin`. If you created an alternative TSM-authorized group during installation, then substitute your group name for `tsmadmin` in the following code examples.

To view the user accounts in the `tsmadmin` group, run the following command:

```bash
 grep tsmadmin /etc/group
```

To add a user account to the `tsmadmin` group:
sudo usermod -G tsmadmin -a <username>

Scripting and automating with tsm CLI

To run automation on a Tableau Server without a password in the script file, run the script on the initial node and with an account in the proper TSM-authorized group. See the "Authenticating" section above for more details.

Viewing help content in the shell

To view minimal help content from a command line, use the `tsm help category`.

Synopsis

`tsm help [category] [command]`

Commands

`tsm help`

Help for all tsm commands

`tsm help <category>`

Show help for a specific command category. For example, `tsm help authentication`.

`tsm help <category> <command>`

Show help for a specific command. For example, `tsm help authentication open-id`.

`tsm help command`

List all top-level commands or categories.
Categories

tsm authentication

You can use the tsm authentication commands to enable, disable, and configure user authentication options for Tableau Server.

- **kerberos**
  - configure
  - disable
  - enable
- **list**
- **mutual-ssl**
  - configure
  - disable
  - enable
  - get-redirect-url
  - map-claims
- **openid**
  - configure
  - disable
  - enable
  - get-redirect-url
  - map-claims
- **saml**
  - configure
  - disable
  - enable
  - export-metadata
  - map-assertions
- **sitesaml**
  - disable
  - enable
- **sspi**
  - disable
  - enable
tsm authentication kerberos <commands>

Enable, disable, and configure Kerberos user authentication on Tableau Server. See Configure Kerberos.

Synopsis

```
 tsm authentication kerberos configure --keytab-file <keytab_file.keytab> [global options]
```

```
 tsm authentication kerberos enable [global options]
```

```
 tsm authentication kerberos disable [global options]
```

Options for kerberos configure

```
-kt, --keytab-file <keytab_file.keytab>
```

Required.

Specifies the service .keytab file used for requests to the KDC.

```

``` tsm authentication list

List the server’s existing authentication-related configuration settings.

Synopsis

```
 tsm authentication list [--verbose][global options]
```

Options

```
 v, --verbose
```

Optional.
Show all configured parameters.

**tsm authentication mutual-ssl <commands>**

Enable, disable, and configure mutual SSL for user authentication on Tableau Server. To learn more about mutual SSL, see Configure Mutual SSL Authentication.

Before you enable mutual SSL, you must enable and configure SSL for external communication. For information, see Configure SSL for External HTTP Traffic to and from Tableau Server.

**Synopsis**

```plaintext
tsm authentication mutual-ssl configure [options] [global options]
tsm authentication mutual-ssl disable [global options]
tsm authentication mutual-ssl enable [global options]
```

**Options**

```plaintext
-cf, --ca-cert <certificate-file.crt>
```

Optional.

Specifies the location and file name for the certificate file. The file must be a valid, trusted certificate from a Certificate Authority (for example, Verisign).

```plaintext
-fb, --fallback-to-basic <true | false>
```

Optional.

Specifies whether Tableau Server should accept user name and password for authentication if SSL authentication fails.

Default value is false, to indicate that when configured for mutual SSL, Tableau Server does not allow a connection when SSL authentication fails. However, Tableau Server
accepts username and password authentication from REST API clients, even if this option is set to false.

-m, --user-name-mapping <upn | ldap | cn>

Optional.

Specifies the user name syntax (UPN, LDAP or CN) to retrieve from identity store or directory. The syntax must match the format for Subject or Subject Alternative Name on the user certificate.

-rf, --revocation-file <revoke-file.pem>

Optional.

Specifies the location and file name for the certificate revocation list file. This file can be a .pem or .der file.

tsm authentication openid <commands>

Enable, disable, and configure OpenID Connect (OIDC) user authentication on Tableau Server.

Synopsis

tsm authentication openid configure [options] [global options]
tsm authentication openid disable [global options]
tsm authentication openid enable [global options]
tsm authentication openid get-redirect-url [global options]
tsm authentication openid map-claims [options] [global options]

Options for openid configure

-a, --client-authentication <string>
Optional.

Specifies custom client authentication method for OpenID Connect.

To configure Tableau Server to use the Salesforce IdP, set this value to `client_secret_post`.

-`cs`, `--client-secret <string>`

Optional.

Specifies the provider client secret. This is a token that is used by Tableau to verify the authenticity of the response from the IdP. This value is a secret and should be kept securely.

-`cu`, `--config-url <CONFIG-URL>`

Optional.

Specifies the provider configuration URL. The default value is constructed based on the name of the server (gateway.public.host), and the gateway port, if any(gateway.public.port). In addition, by default the protocol is set to https:// if SSL is enabled for the server.

-`mf`, `--config-file <config-file.json>`

Optional.

Specifies the local path to the static OIDC discovery JSON document.

-`i`, `--client-id <CLIENT-ID>`

Optional.

Specifies the provider client ID that your IdP has assigned to your application.

-`id`, `--ignore-domain <true | false>`
Optional. Default: false

Set this to true if the following are true:

- You are using email addresses as usernames in Tableau Server
- You have provisioned users in the IdP with multiple domain names
- You want to ignore the domain name portion of the email claim from the IdP

Before you proceed, review the user names that will be used as a result of setting this option to true. User name conflicts may occur. In the case of a user name conflict, the risk of information disclosure is high. See Requirements for Using OpenID Connect.

-if, --iframed-idp-enabled <true | false>
Optional. Default: false

Specifies if IdP is allowed inside of an iFrame. The IdP must disable clickjack protection to allow iFrame presentation.

-ij, --ignore-jwk <true | false>
Optional. Default: false

Set this to true if your IdP does not support JWK validation. In this case, we recommend authenticating communication with your IdP using mutual TLS or another network layer security protocol.

-r, --return-url <return-url>

The URL of your server. This is typically the public name of your server, such as "http://example.tableau.com".

-sn, --custom-scope-name <string>
Optional.
Specifies a custom scope user-related value that you can use to query the IdP. See Requirements for Using OpenID Connect.

Options for openid map-claims

Use these options to change the default OIDC claims Tableau Server will use when communicating with your IdP. See Requirements for Using OpenID Connect.

-i, --id <string>

Optional. Default: sub

Change this value if your IdP does not use the sub claim to uniquely identify users in the ID token. The IdP claim that you specify should contain a single, unique string.

-un, --user-name <string>

Optional. Default: email

Change this value to the IdP claim that your organization will use to match user names as stored in Tableau Server.

tsm authentication saml <commands>

Configure Tableau Server to support single-sign on using the SAML 2.0 standard, enable or disable SAML for a site, map assertion attribute names between Tableau Server and the identity provider (IdP).

Available commands

tsm authentication saml configure [options] [global options]
tsm authentication saml disable [options] [global options]
tsm authentication saml enable [options] [global options]
tsm authentication saml export-metadata [options] [global options]
tsm authentication saml map-assertions [options]

tsm authentication saml configure

Configure the SAML settings for the server. Specify the SAML certificate and metadata files, provide additional required information, set additional options.

If you are configuring SAML for the first time or have previously disabled it, you must run this command with tsm authentication saml enable. For more information, see Configure Server-Wide SAML.

Synopsis

```
tsm authentication saml configure [options] [global options]
```

Options

- `-e`, `--idp-entity-id <id>`
  
  Required for initial SAML configuration; otherwise optional. IdP entity ID value.
  
  Typically this is the same as the Tableau Server return URL (specified in the `--idp-return-url` parameter). The entity ID that you enter is used as a base for generating site-specific entity IDs. For example, if you enter the following:

  ```
  http://tableau-server
  ```

  A site configured for SAML might display the following entity ID:

  ```
  http://tableau-server/samlservice/public/sp/metadata?alias=48957410-9396-430a-967c-75bdb6e002a0
  ```

  To find a site’s entity ID, go to the site’s Settings page, and select the Authentication tab. When SAML is enabled, the entity ID is shown under the first step for configuring site-specific SAML, exporting metadata.

  `-r`, `--idp-return-url <idp-return-url>`

  Required for initial SAML configuration; otherwise optional. The SAML return URL
configured in the IdP. This is typically the Tableau Server external URL; for example, https://tableau-server.

Notes

- http://localhost does not work for an external server.
- Adding a trailing slash to the URL (https://tableau-server/) is not supported.

-i, --idp-metadata <idp-metadata.xml>

Required for initial SAML configuration; otherwise optional. Provide the location and name of the XML metadata file you exported from the IdP’s settings.

For example, /var/opt/tableau/tableau_server/data/saml/<metadata-file.xml>

-cf, --cert-file <certificate.crt>

Required for initial SAML configuration; otherwise optional. The location and file name for the x509 certificate file for SAML. For requirements for the certificate file, see SAML Requirements.

For example, /var/opt/tableau/tableau_server/data/saml/<file.crt>

-kf, --key-file <certificate.key>

Required for initial SAML configuration; otherwise optional. Location and name of the key file that goes along with certificate.

For example, /var/opt/tableau/tableau_server/data/saml/<file.key>

-a, --max-auth-age <max-auth-age>

Optional. Default value is 7200 (2 hours).
The maximum number of seconds allowed between a user’s authentication and processing of the AuthNResponse message.

-\(d\), --desktop-access <enable | disable>
  
  Optional. Default value is enable.

  Use SAML to sign in to the server from Tableau Desktop. If single sign-on from Tableau client applications does not work with your IdP, you can set this to disable.

-\(m\), --mobile-access <enable | disable>

  Optional. Default value is enable.

  Allow using SAML to sign in from older versions of Tableau Mobile app. Devices running Tableau Mobile app version 19.225.1731 and higher ignore this option. To disable devices running Tableau Mobile app version 19.225.1731 and higher, disable SAML as a client login option on Tableau Server.

-\(so\), --signout <enable | disable>

  Optional. Enabled by default.

  Enable or disable SAML sign out for Tableau Server.

-\(su\), --signout-url <url>

  Optional. Enter the URL to redirect to after users sign out of the server. By default this is the Tableau Server sign-in page. You can specify an absolute or a relative URL.

Example

tsm authentication saml configure --idp-entity-id https://tableau-server --idp-metadata /var/opt/tableau/tableau_server/data/saml/<metadata.xml> --idp-return-url https://tableau-server --cert-file /var/opt/tableau/tableau_server-
Tableau Server on Linux Administrator Guide

/data/saml/<file.crt> --key-file /var/opt/tableau/tableau_server-
data/saml/<file.key>

tsm authentication saml enable and saml disable

Enable or disable server-wide SAML authentication. In this context, all sites and users that you enable for SAML go through a single identity provider.

Synopsis

tsm authentication saml enable [global options]
tsm authentication saml disable [global options]

tsm authentication saml export-metadata

Export the Tableau Server .xml metadata file that you will use to configure the SAML IdP.

Synopsis

tsm authentication saml export-metadata [options]
[global options]

Options

-f, --file [/path/to/file.xml]

Optional.

Specifies the location and file name in which the metadata will be written. If you don’t include this option, export-metadata saves the file to the current directory, and names it samlmetadata.xml.

-o, --overwrite

Optional.
Overwrites an existing file of the same name specified in `-f`, or of the default name if `-f` is not included. If a file specified in `-f` exists, and `-o` is not included, the command does not overwrite the existing file.

**tsm authentication saml map-assertions**

Maps attributes between the IdP and Tableau Server. Provide the name that the IdP uses for the attribute specified in each argument.

**Synopsis**

```bash
tsm authentication saml map-assertions --user-name <user-name>
[global options]
```

**Options**

- `--user-name <user-name-attribute>`
  
  Optional. The attribute in which the IdP stores the user name. On Tableau Server this is the display name.

- `--email <email-name-attribute>`
  
  Optional. The name of the attribute in which the IdP stores email addresses.

- `--domain <domain-name-attribute>`
  
  Optional. The attribute in which the IdP stores the domain name.

- `--display-name <display-name-attribute>`
  
  Optional. The attribute in which the IdP stores the domain name.

**Example for saml map-assertions**

```bash
tsm authentication saml map-assertions --email=Email --user-name=DisplayName
```
tsm authentication sitesaml enable and sitesaml disable

Set the server to allow or disallow SAML authentication at the site level. Enabling site-specific SAML gives you access to the Settings > Authentication tab in the Tableau Server web UI. The Authentication tab contains the site-specific SAML configuration settings.

Use the sitesaml enable command with saml configure if you haven’t yet configured the server to allow site-specific SAML. For more information, see Configure Site-Specific SAML.

Synopsis

tsm authentication sitesaml enable [global options]

tsm authentication sitesaml disable [global options]

tsm authentication sspi <commands>

This command will only work on Tableau Server on Windows. If you attempt to enable SSPI on Tableau Server on Linux, an error will be returned.

Enable or disable automatic sign-in using Microsoft SSPI.

If you use Active Directory for authentication, you can optionally enable automatic logon, which uses Microsoft SSPI to automatically sign in your users based on their Windows username and password. This creates an experience similar to single sign-on (SSO).

Do not enable SSPI if you plan to configure Tableau Server for SAML, trusted authentication, a load balancer, or for a proxy server. SSPI is not supported in these scenarios.

Synopsis

tsm authentication sspi disable [global options]

tsm authentication sspi enable [global options]

tsm authentication trusted <commands>

Configure trusted authentication (trusted tickets) for user authentication on Tableau Server.
Synopsis

tsm authentication trusted configure [options] [global options]

Options

-th, --hosts <string>

Optional.

Specifies the trusted host names (or IPv4 addresses) of the web servers that will be hosting pages with Tableau content.

For multiple values, enter the names in a comma-separated list where each value is encapsulated in double-quotes.

For example:

    tsm authentication trusted configure -th "192.168.1.101", "192.168.1.102", "192.168.1.103"

    or

    tsm authentication trusted configure -th "webserv1", "webserv2", "webserv3"

-t, --token-length <integer>

Optional.

Determines the number of characters in each trusted ticket. The default setting of 24 characters provides 144 bits of randomness. The value can be set to any integer between 9 and 255, inclusive.

Global options

-h, --help

Optional.
Show the command help.

-p, --password <password>

Required, along with -u or --username if no session is active.
Specify the password for the user specified in -u or --username.
If the password includes spaces or special characters, enclose it in quotes:
--password 'my password'

-s, --server <url_to_tsm>

Optional.
Use the specified address for Tableau Services Manager. The URL must start with https, include port 8850, and use the server name not the IP address, for example https://mytableauhost:8850. If no server is specified, https://<localhost | dnsname>:8850 is assumed.

--trust-admin-controller-cert

Optional.
Use this flag to trust the self-signed certificate on the TSM controller. For more information about certificate trust and CLI connections, see Connecting TSM clients.

-u, --username <user>

Required if no session is active, along with -p or --password.
Specify a user account. If you do not include this option, the command is run using credentials you signed in with.
You can use the `tsm configuration` commands to get, set, and update configuration values.

For specific configuration keys you can set, see `tsm configuration set` Options.

- `get`
- `set`

## tsm configuration get

View the current server configuration and topology.

**Synopsis**

`tsm configuration get --key <config.key> [global options]`

**Option**

- `-k, --key`

  Required.

  Get the current value of the specified configuration key.

## tsm configuration set

Set or import server configuration or topology.

Quotes around the `<config.key>` and the `<config_value>` are optional unless there are spaces, in which case you must use quotes around the key or value.

**Note:** After setting a configuration key value you must apply the pending configuration changes using `tsm pending-changes apply`. Until you do, the new value will not
be used by Tableau or show up in the results of a `tsm configuration get` command. You can view pending changes using `tsm pending-changes list`. For more information, see `tsm pending-changes`.

**Synopsis**

```bash
tsm configuration set --key <config.key> --value <config_value> [global options]
```

**Options**

- `-k, --key <config.key>`
  
  Required.
  
  Configuration key.

- `-v, --value <config_value>`
  
  Required.
  
  Configuration value.

- `-d`
  
  Optional.
  
  Reset the configuration value to its default.

- `-frc, --force-keys`
  
  Optional.
  
  Force a key to be added to configuration even if it did not previously exist.
Global options

-h, --help

Optional.
Show the command help.

-p, --password <password>

Required, along with -u or --username if no session is active.
Specify the password for the user specified in -u or --username.
If the password includes spaces or special characters, enclose it in quotes:
--password 'my password'

-s, --server <url_to_tsm>

Optional.
Use the specified address for Tableau Services Manager. The URL must start with https, include port 8850, and use the server name not the IP address, for example https://mytableauhost:8850. If no server is specified, https://<localhost | dnsname>:8850 is assumed.

--trust-admin-controller-cert

Optional.
Use this flag to trust the self-signed certificate on the TSM controller. For more information about certificate trust and CLI connections, see Connecting TSM clients.

-u, --username <user>

Required if no session is active, along with -p or --password.
Specify a user account. If you do not include this option, the command is run using credentials you signed in with.

**tsm configuration set Options**

You can use the following configuration options with the `tsm configuration set` command.

**Basic Use of tsm configuration set**

Setting a configuration key

```
 tsm configuration set -k <config.key> -v <config_value>
```

After setting a configuration key value you must apply the pending configuration changes using `tsm pending-changes apply`. Until you do, the new value will not be used by Tableau or show up in the results of a `tsm configuration get` command. You can view pending changes using `tsm pending-changes list`. For more information, see `tsm pending-changes`.

Resetting a configuration key to default

To reset a configuration key back to its default value, use the `-d` option:

```
 tsm configuration set -k <config.key> -d
```

Viewing the current value of a configuration key

To see what a configuration key is currently set to, use the `configuration get` command:

```
 tsm configuration get -k <config.key>
```

**Note:** Configuration keys are case-sensitive.
Options

adminviews.disabled

Default value: false

Disables access to the Tableau Administrative views. By default, access to views is enabled (this option is set to “false”).

api.server.enabled

Default value: true

Allows access to the Tableau Server REST API. By default, this functionality is enabled.

auditing.enabled

Default value: true

Allows access to the PostgreSQL (Tableau Server’s own database) historical auditing tables.

backgrounder.externalquerycachewarmup.enabled

Default value: true

Controls the caching of workbook query results after scheduled extract refresh tasks.

backgrounder.externalquerycachewarmup.view_threshold

Default value: 2.0

The threshold for caching workbook query results after scheduled extract refresh tasks. The threshold is equal to the number of views that a workbook has received in the past seven days divided by the number of refreshes scheduled in the next seven days.
The following two `backgrounder` command options determine how long a flow task can run before the flow background task is canceled. These two commands together determine the total timeout value for flow tasks.

`backgrounder.extra_timeout_in_seconds`

**Default value:** 1800

The number of seconds beyond the setting in `backgrounder.querylimit` before a background task is canceled. This setting makes sure that tasks do not hold up subsequent jobs if they are stalled. The setting applies to processes listed in `backgrounder.timeout_tasks`. 1800 seconds = 30 minutes.

`backgrounder.default_timeout.run_flow`

**Default value:** 14400

The number of seconds for a flow run task is canceled.

`backgrounder.failure_threshold_for_run_prevention`

**Default value:** 5

The number of consecutive failures of a subscription, extract, or flow run job before that job is suspended. Suspending continuously failing jobs helps preserve `backgrounder` resources for other jobs. To disable suspension of failing background tasks, set this to -1.

`backgrounder.querylimit`

**Default value:** 7200

Longest allowable time, in seconds, for completing a single extract refresh task or subscription task. 7200 seconds = 2 hours.
**Note:** If a background task reaches this time limit, it may continue to run for an additional several minutes while being canceled.

**backgrounder.notifications_enabled**

**Default value:** true

Controls whether extract refresh and flow run alerts are enabled for all sites on the server. By default alerts are enabled. To disable the alerts for all sites on a server, set this to false.

Extract alerts can be enabled or disabled on a site basis by site administrators in site settings, or at the user level in user settings.

**backgrounder.sort_jobs_by_run_time_history.observable_hours**

**Default value:** -1

Controls the time window used when determining duration of the last full extract job.

Tableau Server can sort full extract refresh jobs so they are executed based on the duration of their "last run," executing the fastest full extract refresh jobs first.

The "last run" duration of a particular job is determined from a random sample of a single instance of the full extract refresh job in last <n> hours. Full extract jobs are then prioritized to run in order from shortest to longest based on their "last" run duration. By default this is sorting is disabled (-1). If enabling this, the suggested value is 36 (hours).

**backgrounder.sort_jobs_by_type_schedule_boundary.heuristics.milliseconds**

**Default value:** 60000

Controls the time window that identifies backgrounder jobs which are determined to have the same scheduled start time.
Tableau Server on Linux Administrator Guide

The backgrounder process orders work that is scheduled at the same time to be executed by job type, running the fastest category of jobs first: Subscriptions, then Incremental Extracts, then Full Extracts.

Jobs are batched to determine which jobs are scheduled at the “same time”. A value 60,000 milliseconds (the default) indicates jobs for schedules starting within a 1 minute window should be classified in the same batch and so are ordered by type within that batch.

backgrounder.subscription_failure_threshold_for_run_prevention

Default value: 5

Determines the number of consecutive subscription failures that must occur before alerting for a condition is suspended. When set to the default of 5, alerting is suspended after 5 consecutive subscription failures. A value of -1 will allow notification email to continue indefinitely. This threshold is server-wide, so applies to all subscriptions defined on the server.

backgrounder.subscription_image_caching

Default value: true

Controls whether backgrounder will cache images that are generated for subscriptions. Cached images do not have to be regenerated each time so caching improves subscription performance. By default image caching is enabled. To disable image caching for all sites on a server, set this to false.

backgrounder.timeout_tasks

Default value: refresh_extracts,
increment_extracts,
subscription_notify,
single_subscription_notify,
check_data_alert,
run_flow,
encrypt_extracts,
decrypt_extracts,
rekey_extracts,
extract_encryption_maintenance

The list of tasks that can be canceled if they run longer than the combined values in backgrounder.querylimit and backgrounder.extra_timeout_in_seconds. The list of tasks is delimited with commas. The default list represents all the possible values for this setting.

crustercontroller.zk_session_timeout_ms

Default value: 300000

The length of time, in milliseconds, that Cluster Controller will wait for the Coordination Service (ZooKeeper), before determining that failover is required.

dataAlerts.checkIntervalInMinutes

Default value: 60

The frequency, in minutes, at which Tableau Server checks to determine if data-alert conditions are true.

(The server also checks whenever extracts related to data alerts are refreshed.)

dataAlerts.retryFailedAlertsAfterCheckInterval

Default value: true

Determines how often Tableau Server rechecks failing data alerts. When set to true, the server rechecks failing alerts at the frequency defined by dataAlerts.checkIntervalInMinutes. When set to false, the server rechecks failing alerts every five minutes, more quickly notifying alert recipients if data conditions have changed, but reducing server performance.

(The server also checks whenever extracts related to data alerts are refreshed.)
dataAlerts.SuspendFailureThreshold

Default value: **350**

Determines the number of consecutive data alert failures that must occur before alerting for a condition is suspended. When set to the default of 350, alerting is suspended after roughly two weeks of alerts. This threshold is server-wide, so applies to any data alert defined on the server.

DataServerRefreshMetadataPerSession

Default value: **false**

Determines whether Tableau Server will make additional queries to get updated schema data for a published data source when there have been changes in the underlying schema structure. This is disabled by default for performance reasons, and there is a delay in the display of schema changes. If you want changes in the schema of a live published data source to be reflected quickly, or if you see errors (for example, "An error occurred while communicating with the data source: Invalid column name. Statement could not be prepared.") set this to **true**. When set to **true**, Tableau Server makes additional queries to update the schema.

elasticserver.vmopts

Default value: **-Xmx1g -Xms1g**

**Note:** The default value varies based on the size of the memory. If the memory on the machine is 32 GB or higher, the default is set to 1 GB, if the memory on the machine is lower than 32 GB, then the default is set to 256 MB.

Controls the Elastic Server heap size. Tuning these values may help to improve performance. The heap size should usually be less than half of the full machine memory. As a general rule, set initial heap size (-Xms) equal to the maximum heap size (-Xmx) to minimize garbage collections. The default size for these values is measured in...
Here is a suggestion of how much memory to allocate, based on the number of data sources and available memory. Actual performance will vary depending on the server, the number of fields in your data sources, and other factors.

- 0 to 100 data sources: 256 (minimum)
- 100 to 500 data sources: 1 GB (recommended)
- 500 to 1,000 data sources: 2 GB
- 1,000 to 2,000 data sources: 4 GB
- 2,000 to 4,000 data sources: 8 GB
- 4,000 to 8,000 data sources: 16 GB
- 8,000 or more data sources: 32 GB

**excel.shadow_copy_all_remote.enabled**

**Default value:** false

Controls whether Tableau Server creates a "shadow copy" of a shared Excel spreadsheet (.xlsx or .xlsm) that is being used as a live data source. When enabled, this option prevents Excel users from seeing a "Sharing Violation Error" and a message that the file is "currently in use." This option can have a performance impact with large Excel files. If Excel users do not need to edit the shared file, you do not need to enable this option.

**Note:** Tableau Server always attempts to create a shadow copy of a .xls file. This option does not change that behavior.

This option was added beginning with Tableau Server versions: 2019.1.5, 2019.2.1.
Tableau Server on Linux Administrator Guide

Controls whether Tableau Server uses the Apache ActiveMQ service (Tableau Server Messaging Service) for the internal messaging mechanism.

This option was added beginning with Tableau Server version: 2019.4.

features.DesktopReporting

Default value: false

Controls whether Desktop License Reporting is enabled on the server. When set to false (the default), no Administrative Views related to desktop licenses are available. Set this to true to enable license reporting and make license usage and expiration Administrative Views visible on the Server Status page.

features.MessageBusEnabled

Default value: true

Controls whether Tableau Server uses the new internal messaging mechanism.

This option was added beginning with Tableau Server version: 2019.4.

features.PasswordlessBootstrapInit

Default value: true

Controls whether Tableau Server allows embedded credentials in bootstrap files. When enabled (the default), embedded credentials are included in the bootstrap file unless you specify that they should not be included. Set this to false if credentials should never be included in any bootstrap file you generate. For more information on generating bootstrap files, see tsm topology nodes get-bootstrap-file.

This option was added beginning with Tableau Server version 2019.3.

features.PasswordReset

Default value: false
Applies only to servers that use local authentication. Set to true to let users reset their passwords with a "Forgot password" option on the sign-in page.

gateway.http.cachecontrol.updated

Default value: false

The Cache-Control HTTP header specifies whether the client browser should cache content sent from Tableau Server. To disable caching of Tableau Server data on the client, set this option to true.

gateway.http.hsts

Default value: false

The HTTP Strict Transport Security (HSTS) header forces browsers to use HTTPS on the domain where it is enabled.

gateway.http.hsts_options

Default value: "max-age=31536000"

By default, HSTS policy is set for one year (31536000 seconds). This time period specifies the amount of time in which the browser will access the server over HTTPS.

gateway.http.request_size_limit

Default value: 16380

The maximum size (bytes) of header content that is allowed to pass through the Apache gateway on HTTP requests. Headers that exceed the value set on this option will result in browser errors, such as HTTP Error 413 (Request Entity Too Large) or authentication failures.

A low value for gateway.http.request_size_limit can result in authentication errors. Single sign-on solutions that integrate with Active Directory (SAML and Ker-
bers) often require large authentication tokens in HTTP headers. Be sure to test HTTP authentication scenarios before deploying into production.

We recommend setting `tomcat.http.maxrequestsize` option to the same value that you set for this option.

gateway.http.x_content_type_nosniff

Default value: `true`

The X-Content-Type-Options response HTTP header specifies that the MIME type in the Content-Type header should not be changed by the browser. In some cases, where MIME type is not specified, a browser may attempt to determine the MIME type by evaluating the characteristics of the payload. The browser will then display the content accordingly. This process is referred to as "sniffing." Misinterpreting the MIME type can lead to security vulnerabilities. The X-Content-Type-Options HTTP header is set to 'nosniff' by default with this option.

gateway.http.x_xss_protection

Default value: `true`

The HTTP X-XSS-Protection response header is sent to the browser to enable cross-site scripting (XSS) protection. The X-XSS-Protection response header overrides configurations in cases where users have disabled XSS protection in the browser. The X-XSS-Protection response header is enabled by default with this option.

gateway.public.host

Default value: `<hostname>`

The name (URL) of the server, used for external access to Tableau Server. If Tableau Server is configured to work with a proxy server or external load balancer, it is the name entered in a browser address bar to reach Tableau Server. For example, if Tableau Server is reached by entering `tableau.example.com`, the name for `gateway.public.host` is `tableau.example.com`.
gateway.public.port

Default value: 80 (443 if SSL)

Applies to proxy server environments only. The external port the proxy server listens on.

gateway.slow_post_protection.enabled

Default value: false

Enabling this can provide some help in protecting against slow POST (Denial-of-Service) attacks by timing out POST requests that transfer data at extremely slow rates. **Note:** This will not eliminate the threat of such attacks, and could have the unintended impact of terminating slow connections.

gateway.slow_post_protection.request_read_timeout

Default value: header=15-20,MinRate=500 body=10,MinRate=500

When enabled by the preceding option, gateway.slow_post_protection.enabled, this option sets the Apache httpd ReadRequestTimeout. The httpd directive is documented at Apache Module mod_reqtimeout. The primary use of this option is as a defense the Slowloris attack. See the Wikipedia entry, Slowloris (computer security).

gateway.timeout

Default value: 1800

Longest amount of time, in seconds, that the gateway will wait for certain events before failing a request. 1800 seconds = 30 minutes.

gateway.trusted

Default value: IP address of proxy server machine
Applies to proxy server environments only. The IP address(es) or host name(s) of the proxy server.

**gateway.trusted_hosts**

Default value: Alternate names of proxy server

Applies to proxy server environments only. Any alternate host name(s) for the proxy server.

**graph-letingestor.-providerEventIngestorClient.connectionPool.maxConnectionPerInstance**

Default value: 2

Use this option to increase the number of threads allocated to an ingestor instance in the non-interactive microservices container. Ingestor instances are used to perform initial ingestion (index) of content (workbooks, published data sources, and flows) and build relationships between the content and external assets (tables and databases) for Tableau Catalog. Tableau strongly recommends that you progressively increase thread count by only 2 threads at a time while closely monitoring your Tableau Server environment to avoid issues with CPU utilization of the Tableau Sever repository (PostgreSQL database). For more information, see Step 3 (optional): Decrease the time of ingestion and increase memory.

You can use the following command to increase the thread count from the default to 4 threads:

```bash
tsm configuration set -k graph-letingestor.-providerEventIngestorClient.connectionPool.maxConnectionPerInstance -v 4
```
**Important:** When increasing the number of threads allocated to an ingestor instance, you should consider increasing the available memory to support the services associated with Catalog using the `noninteractivecontainer.vmopts` option.

**hyper.file_partition_size_limit**

Default value: 0

When set to 0, the size is set to unlimited and will use all the disk space that is available.

This option is used to set the disk space limit for a query that spools to disk. If your disk space usage by the `spool.<id>.tmp` file is higher than where you need it to be for your environment, it means that queries are spooling and taking up disk space. Use this option to limit the amount of disk space that any one query can use. The `spool.<id>.tmp` file can be found in the temp folder of the user account running Tableau Server. You can specify this value in K(KB), M(MB), G(GB), or T(TB) units. For example, you can specify the size limit as 100G when you want to limit the disk space usage to 100 GB.

For more information about spooling see the Memory and CPU Usage section in Tableau Server Data Engine.

**hyper.global_file_partition_size_limit**

Default value: 0

When set to 0, the size is set to unlimited and will use all the disk space that is available.

This option is used to set the disk space limit for all queries that spool to disk. If your disk space usage by the `spool.<id>.tmp` file is higher than where you need it to be for your environment, it means that queries are spooling and taking up disk space. The `spool.<id>.tmp` file can be found in the temp folder of the user account running Tableau Server. Use this option to limit the amount of disk space in sum total that all queries use when spooling to disk. You can specify this value in K(KB), M(MB), G
You can specify the size limit as 100G when you want to limit the disk space usage to 100 GB. Tableau recommends that you start with this configuration when fine tuning your spooling limits.

For more information about spooling see the Memory and CPU Usage section in Tableau Server Data Engine.

**hyper.log_queries**

Default value: `true`

When set to true, query information is logged.

By default query information is logged. If however you find that the log files are too large for the amount of disk space available, you can set it to `false` to disable logging query information. Tableau recommends leaving this configuration set to `true`.

**hyper.log_query_cpu**

Default value: `false`

Use this setting to log how much time each query takes and the CPU usage.

**hyper.log_timing**

Default value: `false`

This setting is useful to find out more information about the queries, like compilation and parsing times. By default this setting is disabled. You can turn this by setting the value to `true` to collect more details about your queries. Note, however that this will increase the size of your data engine log files (`logs\hyper`).

**hyper.log_troublesome_query_plans**

Default value: `true`
When set to `true`, logs query plans of query that are identified as problematic. Queries that are either canceled, running slower than 10 seconds, or if the queries are spooling to disk fall into this category. The information in the logs can be useful to troubleshoot problematic queries. You can change the setting to `false` if you are concerned about the size of the logs.

**hyper.hard_concurrent_query_thread_limit**

Default value: 100%

Use this option to set the maximum number of threads Data Engine should use for running queries. Since this is a hard limit, use this to when you want to set a hard limit on the CPU usage. This setting will make sure that Data Engine does not use more CPU than this set limit.

It is important to consider that this setting controls the number of concurrent queries that can be executed. So, if you decrease this setting, the chance of queries needing to wait in a queue and wait for currently running queries to complete increases, which may affect workbook load times.

**hyper.soft_concurrent_query_thread_limit**

Default value: 100%

Use this option to specify the number of threads that a single query can be parallelized across, which is the set limit minus the number of active threads already in use. To illustrate this, here is a simplified example:

Let’s say you set this value to 10 threads, this means queries can be parallelized up to 10 threads. If only 2 queries are running, the remaining 8 threads are used to parallelize the 2 queries.

This soft limit makes sure that the Data Engine does not utilize CPU over that limit, unless there are queries waiting to be queued even when at the limit. This is useful when you have queries that are CPU intensive, you can parallelize these across multiple threads so they complete faster.
The `hyper.hard_concurrent_query_thread_limit` and `hyper.soft_concurrent_query_thread_limit` options work together to give you some options to manage your CPU usage while maximizing available CPU resources to complete queries faster. If you don’t want the Data Engine to use all the available CPU on the machine, change it to less than 100% to a percentage that is optimal for your environment. The soft limit is a way for you to limit CPU usage but allow it to go beyond the soft limit up to the hard limit if necessary.

**Note:** The `hyper.hard_concurrent_query_thread_limit` and `hyper.soft_concurrent_query_thread_limit` options replace `hyper.num_job_worker_threads` and `hyper.num_task_worker_threads` options available in Tableau Server versions 2018.3 and earlier, and are deprecated in the current version. For information on the `hyper.num_job_worker_threads` and `hyper.num_task_worker_threads`, see `tsm configuration set Options`.

### hyper.use_spooling_fallback

**Default value:** `true`

When set to `true`, it allows spooling to disk when querying extracts exceeds set RAM usage (80% of installed RAM).

Tableau recommends that you use the default setting. You can turn this off by setting the value to `false` if you are concerned about disk usage. If you turn this setting off, queries that use more than 80% of installed RAM will be canceled.

For more information about spooling see the Memory and CPU Usage section in Tableau Server Data Engine.

### install.firewall.allowedprograms.manage

**Default value:** `true`
Controls whether Tableau Server can add firewall rules. When set to `true` (the default), Tableau Server will add new firewall rules to allow its processes to make connections through Windows Firewall. Change this to `false` if you want to manage all firewall rules yourself and do not want Tableau Server to add new rules.

`java.heap.size`

Default value: 128m

Size of heap for Tomcat (repository and solr). This generally does not need to change except on advice from Tableau.

`licensing.login_based_license_management.default_requested_duration_seconds`

Default value: 0

Set to the duration (in seconds) that a user's login-based license should last before they are prompted to activate Tableau again.

`licensing.login_based_license_management.enabled`

Default value: false

Set to `true` to enable login-based license management. Set to `false` to disable login-based license management.

`licensing.login_based_license_management.max_requested_duration_seconds`

Default value: 15552000

Set to the maximum duration (in seconds) that a user's login-based license should last before they are prompted to activate Tableau again. The maximum value is 15552000 seconds (180 days).

`maestro.input.allowed_paths`
Default value:"

By default, access to any directory will be denied, and only publishing to Tableau Server with content that is included in the tflx file is allowed.

A list of allowed network directories for flow input connections. You must enable Tableau Prep Conductor to run flows on your Tableau Server. For more information, see Tableau Prep Conductor.

The following rules apply and must be considered when configuring this setting:

- Paths should be accessible by Tableau Server. These paths are verified during server startup and at flow run time.

- Network directory paths have to be absolute and cannot contain wildcards or other path traversing symbols. For example `\\myhost\myShare\*` or `\\myhost\myShare*` are invalid paths and would result in all the paths as disallowed. The correct way to safelist any folder under `myShare` would be `\\myhost\myShare` or `\\myhost\myShare/`.

  **Note:** The `\\myhost\myShare` configuration will not allow `\\myhost\myShare1`. In order to safe list both of these folders one would have safe list them as `\\myhost\myShare`; `\\myhost\myShare1`.

- The value can be either `*` meaning that any path, including local (with the exception of some system paths configured using "native_api.internal_disallowed_paths"), or a list of paths, delimited by ";".

  **Note:** If a path is both on the flows allowed list and internal_disassallowed list, internal_disallowed takes precedence.

Important:
This command overwrites existing information and replaces it with the new information you provided. If you want to add a new location to an existing list, you must provide a list of all the
locations, existing and the new one you want to add. Use the following commands to see the current list of input and output locations:

```bash
  tsm configuration get -k maestro.input.allowed_paths
  tsm configuration get -k maestro.output.allowed_paths
```

For more information and details about configuring allowed directories for flow input and output connections, see Safe list Input and Output Locations.

`maestro.output.allowed_paths`

Default value: ""

By default, access to any directories will be denied.

A list of allowed network directories for flow output connections. You must enable Tableau Prep Conductor to run flows on your Tableau Server. For more information, see Tableau Prep Conductor.

The following rules apply and must be considered when configuring this setting:

- Paths should be accessible by Tableau Server. These paths are verified during server startup and at flow run time.

- Network directory paths have to be absolute and cannot contain wildcards or other path traversing symbols. For example `\\myhost\myShare\*` or `\\myhost\myShare*` are invalid paths and would result in all the paths as disallowed. The correct way to safelist any folder under `myShare` would be `\\myhost\myShare` or `\\myhost\myShare\`.

  **Note:** The `\\myhost\myShare` configuration will not allow `\\myhost\myShare1`. In order to safe list both of these folders one would have safe list them as `\\myhost\myShare`; `\\myhost\myShare1`.

- The value can be either `*` meaning that any path, including local (with the
exception of some system paths configured using "native_api.internal_disallowed_paths"), or a list of paths, delimited by ",".

**Note:** If a path is both on the flows allowed list and internal_disallowed list, internal_disallowed takes precedence.

For more information and details about configuring allowed directories for flow input and output connections, see Safe list Input and Output Locations.

**metadata.query.limits.time**

Default value: 20

This is the longest allowable time, in seconds, for a Catalog or Metadata API query to run before a timeout occurs and the query is canceled. Tableau recommends incrementally increasing the timeout limit to *no more than* 60 seconds using the following command:

```
tsm configuration set -k metadata.query.limits.time -v PT30S --force-keys
```

**Important:** This option should be changed only if you see the error described here, Timeout limit and node limit exceeded messages. Increasing the timeout limit can utilize more CPU for longer, which can impact the performance of tasks across Tableau Server. Increasing the timeout limit can also cause higher memory usage, which can cause issues with the interactive microservices container when queries run in parallel.

**metadata.query.limits.count**

Default value: 2000

This is the number of objects (which can loosely map to the number of query results) that Catalog can return before the node limit is exceeded and the query is canceled. Tableau recommends incrementally increasing the timeout limit, to *no more than* 100,000 using the following command:
tsm configuration set -k metadata.query.limits.count -v 3000 --force-keys

**Important:** This option should be changed only if you see the error described here, Timeout limit and node limit exceeded messages. Increasing the node limit can cause higher memory usage, which can cause issues with the interactive microservices container when queries run in parallel.

monitoring.dataengine.connection_timeout

**Default value:** 30000

The length of time, in milliseconds, that Cluster Controller will wait for the data engine, before determining that a connection timeout occurred. The default is 30,000 milliseconds (30 seconds).

native_api.connection.limit.<connection class>

Set parallel query limit for the specified data source (connection class). This overrides the global limit for the data source.

native_api.connection.globallimit

**Default value:** 16

Global limit for parallel queries. Default is 16 except for Amazon Redshift which has a default of 8.

native_api.ProtocolTransitionLegacyFormat

**Default value:** false

Use the legacy name format for constrained delegation.

The name format was changed in version 10.1 to allow cross-domain protocol transition (S4U). If this causes problems with existing configurations and you don’t need
cross-domain protocol transition, configure Tableau Server to use the old behavior by setting this to true.

native_api.unc_mountpoints

Default value: none

Specifies UNC and FQDN path for shared Windows directories that are accessed by Tableau Server on Linux. Each path must also be referenced in a corresponding auto.\-cifs file. Separate each path by a semicolon, for example:

'//files-rv01/development;/mnt/filesrv01/development;//filesrv01.example.lan/development;/mnt/filesrv01/development'

Subsequent updates to the native_api.unc_mountpoints value will overwrite the existing value. Therefore, each time you add a Windows share, you must include all shares in the updated value.

For more information, see the Community wiki topic, Connecting to a Windows Shared Directory.

noninteractivecontainer.vmopts

Default value: Xmx64m \(-Xms4g\)

Use this option to increase the JVM heap size for Tableau Catalog ingestion. This value should only be increased if the number of threads allocated to the non-interactive microservices container is being increased using the graph-letingestor.providerEventIngestorClient.connectionPool.maxConnectionPerInstance option. Tableau strongly recommends adding no more than 2 GB of memory per additional thread count added. For more information, see Step 3 (optional): Decrease the time of ingestion and increase memory.

You can use the following command to increase the max heap size from the default to 6 GB:
tsm configuration set -k noninteractive.vmopts -v "-XX:+UseConcMarkSweepGC -Xmx6g -Xms64m -XX:+ExitOnOutOfMemoryError"

pgsql.port

Default value: 8060

Port that PostgreSQL listens on.

pgsql.preferred_host

Specifies the computer name of the node with the preferred repository installed. This value is used if the --preferred or -r option is specified with the tsm topology fail-over-repository command.

Example:

tsm configuration set -k pgsql.preferred_host -v "<host_name>"

Note: The host_name is case-sensitive and must match the node name shown in the output of tsm status -v.

pgsql.verify_restore.port

Default value: 8061

Port used to verify the integrity of the PostgreSQL database. See tsm maintenance backup for more information.

recommendations.enabled

Default value: true

Controls the recommendations feature, which powers recommendations for data sources and tables (for Tableau Desktop) and recommendations for views (for
Tableau Server on Linux Administrator Guide

Tableau Server). Recommendations are based on the popularity of content and on content used by other users determined to be similar to the current user.

`recommendations.vizrecs.enabled`

Default value: true

Controls recommendations for views for Tableau Server users. This option is a child of `recommendations.enabled` and will have no effect if the parent option is set to false. When the parent option is set to true, and this option is set to false, data sources and tables will still be recommended to Tableau Desktop users, but recommendations for views on Tableau Server will be disabled.

`refresh_token.absolute_expiry_in_seconds`

Default value: 31536000

Specifies the number of seconds for absolute expiry of OAuth refresh and access tokens. The OAuth tokens are used by clients for authentication to Tableau Server after initial sign-in. To remove limits set to -1. To disable OAuth tokens, see Disable Automatic Client Authentication.

`refresh_token.idle_expiry_in_seconds`

Default value: 1209600

Specifies the number of seconds when idle OAuth tokens will expire. The OAuth tokens are used by clients for authentication to Tableau Server after initial sign-in. To remove limits set to -1.

`refresh_token.max_count_per_user`

Default value: 24
Specifies the maximum number of refresh tokens that can be issued for each user. If user sessions are expiring more quickly than you expect, either increase this value or set it to −1 to entirely remove token limits.

rsync.timeout

Default value: 600

Longest allowable time, in seconds, for completing file synchronization (600 seconds = 10 minutes). File synchronization occurs as part of configuring high availability, or moving the data engine and repository processes.

schedules.display_schedule_description_as_name

Default value: false

Controls whether a schedule name displays when creating a subscription or extract refresh (the default), or the "schedule frequency description" name describing the time and frequency of the schedule displays. To configure Tableau Server to display timezone-sensitive names for schedules, set this value to true.

When true, the "schedule frequency description" is also displayed after the schedule name on the schedule list page.

schedules.display_schedules_in_client_timezone

Default value: true

Shows the "schedule frequency description" in the timezone of the user when true (uses the client browser timezone to calculate the "schedule frequency description").

service.jmx_enabled

Default value: false

Setting to true enables JMX ports for optional monitoring and troubleshooting.
service.max_procs

Default value: <number>

Maximum number of server processes.

service.port_remapping.enabled

Default value: true

Determines whether or not Tableau Server will attempt to dynamically remap ports when the default or configured ports are unavailable. Setting to false disables dynamic port remapping.

session.ipsticky

Default value: false

Makes client sessions valid only for the IP address that was used to sign in. If a request is made from an IP address different from that associated with the session token, the session token is considered invalid.

In certain circumstances—for example, when Tableau Server is being accessed by computers with known and static IP addresses—this setting can yield improved security.

**Note:** Consider carefully whether this setting will help your server security. This setting requires that the client have a unique IP address and an IP address that stays the same for the duration of the session. For example, different users who are behind a proxy might look like they have the same IP address (namely, the IP address of the proxy); in that case, one user might have access to another user’s session. In other circumstances, users might have a dynamic IP address, and their address might change during the course of the session. If so, the user has to sign in again.
**sheet_image.enabled**

Default value: true

Controls whether you can get images for views with the REST API. For more information, see REST API Reference.

**solr.rebuild_index_timeout**

Default value: 7200

When Tableau Server is upgraded or when a .tsbak file is restored, the background task rebuilds the search index. This setting, in seconds, controls the timeout setting for that task (7200 seconds = 120 minutes).

**storage.monitoring.email_enabled**

Default value: false

Controls whether email notifications are enabled for server disk space monitoring. By default, email notifications are enabled. To enable notifications for disk space monitoring, set this to true.

SMTP must be configured for notifications to be sent. For details, see Configure SMTP Setup.

**storage.monitoring.warning_percent**

Default value: 20

Warning threshold of remaining disk space, in percentage of total disk space. If disk space falls below this threshold, a warning notification is sent.

**storage.monitoring.critical_percent**

Default value: 10
Critical threshold of remaining disk space, in percentage of total disk space. If disk space falls below this threshold, a critical notification is sent.

storage.monitoring.email_interval_min

Default value: 60

How often, in minutes, that email notifications should be sent when disk space monitoring is enabled and a threshold is crossed.

storage.monitoring.record_history_enabled

Default value: true

Determines whether free disk space history is saved and available to view in Administrative Views. To disable history storage for monitoring, set storage.monitoring.record_history_enabled to false.

subscriptions.enabled

Default value: false

Controls whether subscriptions are configurable system-wide. See Set Up a Site for Subscriptions.

subscriptions.timeout

Default value: 1800

Longest allowable time, in seconds, for a single view in a workbook subscription task to be rendered before the task times out. This value applies separately to each view in the workbook, so the total length of time to render all the views in a workbook (the full subscription task) may exceed this timeout value. 1800 seconds = 30 minutes.

svcmonitor.notification.smtp.enabled

Default value: false
Controls whether email notifications are enabled for server process events. By default notifications are sent when processes go down, fail over, or restart. To enable server process notifications, set this to true.

SMTP must be configured for notifications to be sent. For details, see Configure SMTP Setup.

tabadmincontroller.auth.expiration.minutes

Default value: 120

Controls how long session cookies are valid. By default this is set to 120 minutes. This value also determines how long the embedded credentials in a node bootstrap file are valid. For more information, see tsm topology nodes get-bootstrap-file.

tomcat.http.maxrequestsize

Default value: 16380

The maximum size (bytes) of header content that is allowed to pass through the Apache gateway on HTTP requests. Headers that exceed the value set on this option will result in browser errors, such as HTTP Error 413 (Request Entity Too Large) or authentication failures.

A low value for tomcat.http.maxrequestsize may result in authentication errors. Single sign-on solutions that integrate with Active Directory (SAML and Kerberos) often require large authentication tokens in HTTP headers. Be sure to test HTTP authentication scenarios before deploying into production.

We recommend setting gateway.http.request_size_limit option to the same value that you set for this option.

tomcat.https.port

Default value: 8443

SSL port for Tomcat (unused).
**tomcat.server.port**

Default value: 8085

Port that tomcat listens on for shutdown messages.

**vizportal.adsync.update_system_user**

Default value: false

Specifies whether email addresses and display names of users are changed (even when changed in Active Directory) when an Active Directory group is synchronized in Tableau Server. To ensure that user email addresses and display names are updated during synchronization, set `vizportal.adsync.update_system_user` to true, and then restart the server.

**vizportal.commenting.delete_enabled**

Default value: true

When set to true, lets users delete comments on views. You can delete a comment if you created it, are the content owner, a project leader with an appropriate site role, or are an administrator. To learn which site roles are required for full project leader access, see Project-level administration.

**vizportal.csv_user_mgmt.index_site_users**

Default value: true

Specifies whether indexing of site users is done user by user when importing or deleting users with a CSV file. When set to true (the default) indexing is done as each user is added or deleted. To delay the indexing of the site users until after the entire CSV file has been processed, set this to false.

**vizportal.log.level**

Default value: info
The logging level for vizportal Java components. Logs are written to /var/opt/tableau/tableau_server/data/tabsvc/logs/vizportal/*.log.

Set to debug for more information. Using the debug setting can significantly impact performance, so you should only use this setting when directed to do so by Tableau Support.

vizportal.openid.client_authentication

Specifies custom client authentication method for OpenID Connect.

To configure Tableau Server to use the IdPs that require the client_secret_post, set this value to client_secret_post.

An example would be when connecting to the Salesforce IDP, which requires this.

vizportal.rest_api.cors.allow_origin

Specifies the origins (sites) that are allowed access to the REST API endpoints on Tableau Server when vizportal.rest_api.cors.enabled is set to true. You can specify more than one origin by separating each entry with a comma (,).


If vizportal.rest_api.cors.enabled is false, the origins listed by this option are ignored. For more information, see Enabling CORS on Tableau Server.

Note: You can use an asterisk (*) as a wild card to match all sites. This is not recommended as it allows access from any origin that has access to the server and can present a security risk. Do not use an asterisk (*) unless you fully understand the implications and risks for your site.

vizportal.rest_api.cors.enabled

Default value: false
Tableau Server on Linux Administrator Guide

Controls whether Tableau Server allows Cross Origin Resource Sharing (CORS). When set to `true`, the server allows web browsers to access the Tableau REST API endpoints. You can use this option and the REST API to create custom portals. By default, this functionality is not enabled. To specify which origins (sites) have access, use the `vizportal.rest_api.cors.allow_origin` option. Only the origins specified with this option are allowed to make requests to the Tableau Server REST API. For more information, see Enabling CORS on Tableau Server.

`vizportal.rest_api.view_image.max_age`

Default value: 1

The amount of time, in minutes, to cache images that are generated by the Query View Image method of the REST API. For more information, see the REST API Reference in the REST API help.

`vizqlserver.allow_insecure_scripts`

Default value: `false`

Allows a workbook to be published to the server from Tableau Desktop, and to be opened from the server, even if the workbook contains SQL or R expressions that are potentially unsafe (for example, a SQL expression that could potentially allow SQL injection). When this setting is `false` (the default), publishing a workbook or opening it from the server results in an error message, and the workbook is blocked. Before you set this value to `true` review the Knowledge Base article, Blocking or Allowing Insecure Scripts in Tableau Server.

`vizqlserver.browser.render`

Default value: `true`

Views under the threshold set by `vizqlserver.browser.render_threshold` or `vizqlserver.browser.render_threshold_mobile` are rendered by the client web browser instead of by the server. See Configure Client-Side Rendering for details.
vizqlserver.browser.render_threshold

Default value: 100

The default value represents a high level of complexity for a view displayed on a PC. Complexity factors include number of marks, headers, reference lines, and annotations. Views that exceed this level of complexity are rendered by the server instead of in the PC's web browser.

vizqlserver.browser.render_threshold_mobile

Default value: 60

The default value represents a high level of complexity for a view displayed on a tablet. Complexity factors include number of marks, headers, reference lines, and annotations. Views that exceed this level of complexity are rendered by the server instead of in the tablet's web browser.

vizqlserver.clear_session_on_unload

Default value: false

Determines whether or not VizQL sessions are kept in memory when a user navigates away from a view or closes their browser. The default value (false) keeps sessions in memory. To close VizQL sessions on leaving a view or closing a browser, set this to true.

vizqlserver.geosearch_cache_size

Default value: 5

Sets the maximum number of different geographic search locale/language data sets that can be loaded into server memory at the same time. When the server receives a geographic search request for locale/language data set that is not in memory, it will load the set into memory. If loading the data set will exceed the specified limit, the least recently used locale/language data set is cleared from memory so the
Tableau Server on Linux Administrator Guide

requested one can be loaded. The minimum value is 1. Each cache takes approximately 60 MB in memory (so if you set this to 10, the memory usage would be 600 MB (60 * 10).

vizqlserver.initialsql.disabled

Default value: false

Specify whether to ignore initial SQL statements for all data sources. Set this to true to ignore initial SQL:

tsm configuration set -k vizqlserver.initialsql.disabled -v true

vizqlserver.log.level

Default value: info

The logging level for vizportal Java components. Logs are written to /var/opt/tableau/tableau_server/data/tabsvc/logs/vizportal/*.log.

Set to debug for more information. Using the debug setting can significantly impact performance, so you should only use it when directed to do so by Tableau Support.

vizqlserver.NumberOfWorkbookChangesBetweenAutoSaves

Default value: 5

Auto recover configuration for web authoring. Specifies the number of changes that a user must make to trigger auto save. Take care when changing this value. Auto recover functionality may impact the performance of web authoring and other viz-related operations on Tableau Server. We recommend tuning this value by making incremental adjustments over time.

vizqlserver.port

Default value: 9100
Base port for the VizQL servers.

vizqlserver.protect_sessions

Default value: true

When set to true, prevents VizQL sessions from being reused after the original user signs out.

vizqlserver.querylimit

Default value: 1800

Longest allowable time for updating a view, in seconds. 1800 seconds = 30 minutes.

vizqlserver.RecoveryAttemptLimitPerSession

Default value: 3

Auto recover configuration for web authoring. The maximum number of attempts to recover the same session. Take care when changing this value. Auto recover functionality may impact the performance of web authoring and other viz-related operations on Tableau Server. We recommend tuning this value by making incremental adjustments over time.

vizqlserver.session.expiry.minimum

Default value: 5

Number of minutes of idle time after which a VizQL session is eligible to be discarded if the VizQL process starts to run out of memory.

vizqlserver.session.expiry.timeout

Default value: 30

Number of minutes of idle time after which a VizQL session is discarded.
vizqlserver.showDownload

**Default value:** true

Controls the display of the **Tableau Workbook** option of the Download menu in views. When set to false, the Tableau Workbook option is unavailable.

vizqlserver.showShare

**Default value:** true

Controls the display of Share options in views. To hide these options, set to false.

**Note:** Users can override the server default by setting the "showShareOptions" JavaScript or URL parameter.

vizqlserver.url_scheme_whitelist

Specifies one or more URL schemes to whitelist when using **URL actions** on views and dashboards. The schemes http, https, gopher, mailto, news, sms, tel, tsc, and tsl are whitelisted by default. This command can contain multiple comma and space-separated values, as in this example:

```
  tsm configuration set -k vizqlserver.url_scheme_whitelist -v scheme1, scheme2
```

The values you specify overwrite previous settings. Therefore, you must include the full list of schemes in the set command. (You cannot amend the list of schemes by running the set command repeatedly.)

vizqlserver.WorkbookTooLargeToCheckpointSizeKiB

**Default value:** 1024
Auto recover configuration for web authoring. Size limit (KB) for a workbook that will auto save. Workbooks larger than this value will not be auto-saved. Take care when changing this value. Auto recover functionality may impact the performance of web authoring and other viz-related operations on Tableau Server. We recommend tuning this value by making incremental adjustments over time.

webdataconnector.refresh.enabled

Deprecated. Use `tsm data-access web-data-connectors allow` instead.

Determines whether extract refreshes for web data connectors (WDCs) are enabled in Tableau Server. To disable refresh for all WDCs, set the value for this key to `false`, as shown below:

```
tsm configuration set --key webdataconnector.refresh.enabled --value false
```

To learn more, see [Web Data Connectors in Tableau Server](#).

webdataconnector.whitelist.fixed

Deprecated. Use `tsm data-access web-data-connectors add` instead.

Specifies one or more web data connectors (WDCs) that can be used by to access data connections that are accessible over HTTP or HTTPS. This command is formatted as JSON data on a single line, with all double-quotes (") escaped using a back-slash (\).

For example to add a San Francisco Film Locations WDC to the safe list:

```
tsm configuration set --key webdataconnector.whitelist.fixed --value "'{"https://tableau.data.world:443": {"properties": { \"secondary_whitelist\": [\"(https://data.world/)\"] } } }'"
```

To learn more, see [Web Data Connectors in Tableau Server](#).

webdataconnector.enabled
**Deprecation Note:** Use `tsm data-access web-data-connectors allow` instead.

**Default value:** `true`

When set to `true`, you can use `tsm` commands to manage web data connectors on the server.

**webdataconnector.whitelist.mode**

**Default value:** `mixed`

Determines how Tableau Server can run web data connectors. Supported modes are:

- **fixed.** Users can run connectors that are on a safe list (whitelist) of URLs.
- **insecure.** Users can run any connector.

**Important:** Use the `insecure` option *only* for development and testing. Because connectors run custom code, running connectors that have not been vetted can pose a security threat.

**wgserver.audit_history_expiration_days**

**Default value:** `183`

Specifies the number of days after which historical events records are removed from the PostgreSQL database (the Tableau Server database).

**wgserver.change_owner.enabled**

**Default value:** `true`

Controls whether the ownership of a workbook, data source or project can be changed. Other options include `false` and `adminonly`.

**wgserver.clickjack_defense.enabled**

**Default value:** `true`
When set to `true`, helps prevents a malicious person from "clickjacking" a Tableau Server user. In a clickjack attack, the target page is displayed transparently over a second page, and the attacker gets the user to click or enter information in the target page while the user thinks he or she is interacting with the second page.

For more information, see Clickjack Protection.

**wgserver.domain.ldap.domain_custom_ports**

Default value: `null`

Allows you to map child domains and their LDAP ports. Domain and port are separated by a colon (:) and each domain:port pair is separated by a comma (,) using this format: `FQDN1:port,FQDN2:port`


**wgserver.extended_trusted_ip_checking**

Default value: `false`

Enforces IP client matching for trusted ticket requests.

**wgserver.restrict_options_method**

Default value: `true`

Controls whether Tableau Server accepts HTTP OPTIONS requests. If this option is set to `true`, the server returns HTTP 405 (Method Not Allowed) for HTTP OPTIONS requests.

**wgserver.saml.idpattribute.username**

Specifies the name of the attribute in which your SAML IdP stores user names. By default, this is set to `username`. If the attribute name that your IdP uses contains
Tableau Server on Linux Administrator Guide

spaces, enclose it in quotation marks. For more information, see Configure Server-Wide SAML or Configure Site-Specific SAML.

wgserver.saml.iframed_idp.enabled

Default value: false

Default of false means that when users select the sign-in button on an embedded view, the IdP’s sign-in form opens in a pop-up window.

When you set it to true, and a server SAML user who is already signed in navigates to a web page with an embedded view, the user will not need to sign in to see the view.

You can set this to true only if the IdP supports signing in within an iframe. The iframe option is less secure than using a pop-up, so not all IdPs support it. If the IdP sign-in page implements clickjack protection, as most do, the sign-in page cannot display in an iframe, and the user cannot sign in.

If your IdP does support signing in via an iframe, you might need to enable it explicitly. However, even if you can use this option, it disables Tableau Server clickjack protection for SAML, so it still presents a security risk.

wgserver.saml.maxassertiontime

Default value: 3000

Specifies the maximum number of seconds, from creation, that a SAML assertion is usable.

wgserver.saml.responseskew

Default value: 180

Sets the maximum number of seconds difference between Tableau Server time and the time of the assertion creation (based on the IdP server time) that still allows the message to be processed.
**wgserver.session.apply_lifetime_limit**

**Default value:** `false`

Controls whether there is a session lifetime for server sessions. Set this to `true` to configure a server session lifetime.

**wgserver.session.idle_limit**

**Default value:** `240`

The number of minutes of idle time before a sign-in to the web application times out.

**wgserver.session.lifetime_limit**

**Default value:** `1440`

The number of minutes a server session lasts if a session lifetime is set. The default is 1440 minutes (24 hours). If `wgserver.session.apply_lifetime_limit` is `false` (the default) this is ignored.

**wgserver.unrestricted_ticket**

**Default value:** `false`

Specifies whether to extend access to server resources for users authenticated by trusted tickets. Default behavior allows users to access views only. Setting this to `true` allows users with valid trusted tickets to access server resources (projects, workbooks, and so on) as if they had signed in using their credentials.

**workerX.gateway.port**

**Default value:** `80` (**443** if SSL)

External port that Apache listens on for workerX (where a “worker” is the term used for subsequent server nodes in the cluster). `worker0.gateway.port` is Tableau
Server's external port. In a distributed environment, worker0 is the initial Tableau Server node.

workerX.vizqlserver.procs

Default value: <number>

Number of VizQL servers.

zookeeper.config.snapCount

Specifies the number of transactions necessary to cause the Coordination Service to create a snapshot of the logs. By default this value is 100,000 transactions. If your Coordination Service is not writing enough transactions to result in snapshots, the automatic cleanup of snapshots older than five days will not take place, and you may lose disk space to the transaction logs. By default transaction logs and snapshots are created in the Tableau data directory.

**tsm customize**

You can use the tsm customize command to customize the look and feel of the Tableau Server client browser experience.
<table>
<thead>
<tr>
<th>Logo</th>
<th>Option flag</th>
<th>Minimum size/Max-imum size, in pixels</th>
<th>Recommended size, in pixels</th>
</tr>
</thead>
<tbody>
<tr>
<td>A - Window tab logo</td>
<td>cannot be changed</td>
<td>cannot be changed</td>
<td>cannot be changed</td>
</tr>
<tr>
<td>B - Server name</td>
<td>--server-name</td>
<td>does not apply</td>
<td>does not apply</td>
</tr>
<tr>
<td>C - Header logo</td>
<td>--header-logo</td>
<td>32 by 32 min, 160 by 48 max</td>
<td>48 by 48</td>
</tr>
<tr>
<td>Not shown - Sign in logo</td>
<td>--signin-logo</td>
<td>3000 by 3000 max</td>
<td></td>
</tr>
<tr>
<td>Header logo/Sign in logo</td>
<td>--logo</td>
<td>32 by 32 min, 160 by 48 max</td>
<td>48 by 48</td>
</tr>
<tr>
<td>Not shown - Logo shown when navigation pane is minimized</td>
<td>--compact-logo</td>
<td>32 by 32 max</td>
<td>32 by 32</td>
</tr>
</tbody>
</table>
The image files you use should be in GIF, JPEG, or PNG format.

The background colors on the header and sign in page are not the same. If you use the same image for both locations (if you use the `--logo` option, for example) your logo might look different depending on where it appears in the server interface.

Synopsis

```bash
tsm customize [options] [global options]
```

After you run the `customize` command, you must run the following command to apply changes:

```bash
tsm pending-changes apply
```

Options

**Note:** Use quotation marks around the path and image file if they include any spaces.

```bash
--compact-logo "<path-to-logo>"
```

Optional.

Specify a path to the image file that will be displayed when the navigation pane size is minimized. The maximum (and optimal) size is 32 by 32 pixels.

```bash
--header-logo "<path-to-logo>"
```

Optional.

Specify a path to the image file that will be displayed in the header only.

```bash
--logo "<path-to-logo>"
```

Optional.
Path to a single image file that will display for both the header and the sign-in window.

```--restore-defaults```
Optional.
Reset all customization options to default install state.

```--server-name <server_name>```
Optional.
Server name that appears in the browser tab, tooltips, and messages.

```--signin-logo "<path-to-logo>"```
Optional.
Specify a path to the image file that will be displayed for sign-in window only.

Global options

```-h, --help```
Optional.
Show the command help.

```-p, --password <password>```
Required, along with `-u` or `--username` if no session is active.
Specify the password for the user specified in `-u` or `--username`.
If the password includes spaces or special characters, enclose it in quotes:

```--password 'my password'```

```-s, --server <url_to_tsm>```
Optional.

Use the specified address for Tableau Services Manager. The URL must start with https, include port 8850, and use the server name not the IP address, for example https://mytableauhost:8850. If no server is specified, https://<localhost | dnsname>:8850 is assumed.

--trust-admin-controller-cert

Optional.

Use this flag to trust the self-signed certificate on the TSM controller. For more information about certificate trust and CLI connections, see Connecting TSM clients.

-u, --username <user>

Required if no session is active, along with -p or --password.

Specify a user account. If you do not include this option, the command is run using credentials you signed in with.

tsm data-access

You can use the tsm data-access commands to configure data caching, enable or disable data repository access, enable SAML for single sign-on, and configure settings for Web Data Connectors (WDCs).

- caching
  - data-access caching list
  - data-access caching set
- repository
  - repository-access disable
  - repository-access enable
  - repository-access list
- set-saml-delegation
  - set-saml-delegation configure
  - set-saml-delegation disable
  - set-saml-delegation enable
- web-data-connectors
  - web-data-connectors add
  - web-data-connectors allow
  - web-data-connectors delete
  - web-data-connectors list

tsm data-access caching list

Displays data connection caching settings. To learn more about data connection caching settings, see Configure Data Cache.

Synopsis

tsm data-access caching list [global options]

tsm data-access caching set

Sets data connection caching settings. To learn more about data connection caching settings, see Configure Data Cache.

Synopsis

tsm data-access caching set [options] [global options]

Options

- -r, --refresh-frequency

Optional.

Sets the frequency to refresh cached data with a new query to the underlying data source. You can specify a number to define the maximum number of minutes that
data should be cached. You can also specify `low` to cache and reuse data for as long as possible, or `always` (equivalent to 0) to refresh data each time that a page is loaded. If this option is not specified, it defaults to `low`.

**tsm data-access repository-access disable**

Disable external access to the Tableau PostgreSQL database for the default remote user. This will not disable access from localhost.

**Synopsis**

```
 tsm data-access repository-access disable [options]
 [global options]
```

**Options**

---repository-username `<username>`

Required.

The username, either `tableau` or `readonly`, with access to the data repository.

---request-timeout `<timeout in seconds>`

Optional.

Wait the specified amount of time for the command to finish. Default value is 1800 (30 minutes).

---ignore-prompt

Optional.

Suppress the prompt for restart and restart Tableau Server.

**tsm data-access repository-access enable**

Enables access to the Tableau PostgreSQL database.
By default, PostgreSQL traffic uses port 8060 (TCP). If you are running a local firewall, be sure to allow traffic for this port. To change the PostgreSQL port, see Ports that are not dynamically mapped.

Synopsis

tsm data-access repository-access enable [options]
[global options]

Options

--repository-password <password>

Required.

Sets (or changes) the password to access the data repository for the specified username.

--repository-username <username>

Required.

The username, either tableau or readonly, with access to the data repository.

--request-timeout <timeout in seconds>

Optional.

Wait the specified amount of time for the command to finish. Default value is 1800 (30 minutes).

--ignore-prompt

Optional.

Suppress the prompt for restart and restart Tableau Server.
Tableau Server on Linux Administrator Guide

tsm data-access repository-access list

Lists users who have access to the Tableau PostgreSQL database.

Synopsis

```
  tsm data-access repository-access list [global options]
```

tsm data-access set-saml-delegation configure

Setup single sign-on for SAML SAP HANA so that Tableau Server functions as an Identity Provider (IdP) that provides single sign-on for users making SAP HANA data connections.

Synopsis

```
  tsm data-access set-saml-delegation configure [options]
    [global options]
```

Options

- `-kf, --cert-key <cert-key>`
  Optional.
  The SAML certificate key file.

- `-cf, --cert-file <file-path>`
  Optional.
  The location of the SAML certificate file.

- `-uf, --username-format <username-format>`
  Optional.
  Username format. Valid format keys are: 'username', 'domain_and_username', and 'email'.

Tableau Software  Version: 2019.4  1273
-uc, --username-case <username-case>

Optional.

Username case. Valid case keys are: 'lower', 'upper', and 'preserve'.

**tsm data-access set-saml-delegation disable**

Disable single sign-on for SAML SAP HANA.

**Synopsis**

```tsm data-access set-saml-delegation disable [global options]```

**tsm data-access set-saml-delegation enable**

Enable single sign-on for SAML SAP HANA.

**Synopsis**

```tsm data-access set-saml-delegation enable [global options]```

**tsm data-access web-data-connectors add**

Add a web data connector (WDC) to the WDC safe list ("whitelist").

**Synopsis**

```tsm data-access web-data-connectors add [options] [global options]```

**Options**

- `-n, --name <name>`
  
  Required.
The name for the WDC that will be displayed in the Tableau Server data source list. This name must be enclosed in single quotes ('') or double quotes ("). 

-sec, --secondary <secondary-URL-1>, <secondary-URL-2>

Optional.

A comma-delimited list of URLs that indicates which domains the connector can make requests to or receive data from. To add an entire domain to this secondary safe list, you can use a wildcard expression (.*). Be sure to include the parentheses ( ) as part of the expression, as in the following example: https://www.example.com/(.*)

--url <URL>

Required.

The URL for the WDC (formatted as <scheme>://<host>:<port>/<path>, for example https://www.tableau.com:80/example/). For many WDCs, the <port> value is 443 or 80, but you can check the value for your connector by looking at the data source details on Tableau Server or Tableau Online.

tsm data-access web-data-connectors allow

Enable or disable WDC refreshes. Also, enable or disable the use of WDCs on Tableau Server.

Synopsis

tsm data-access web-data-connectors allow [options]
[global options]

Options

-r, --refreshes <refreshes-allowed>

Optional.
Set to **false** to disallow WDC refreshes. Defaults to **true**.

-\(t\), --**type** \(<\text{WDC-allowed}>\)

Optional.

Set to **none** to disallow the use of WDCs on Tableau Server (and omit WDCs from backups). Defaults to **all**, which allows the use of WDCs.

**tsm data-access web-data-connectors delete**

Delete a specified WDC, or all WDCs, from the Tableau Server safe list ("whitelist").

**Synopsis**

`tsm data-access web-data-connectors delete [options]`

[global options]

**Options**

--**all**

Optional.

This option will delete all WDCs.

--**url** \(<\text{URL}>\)

Optional.

The URL for the WDC to delete.

**tsm data-access web-data-connectors list**

List all WDCs currently on the safe list ("whitelist").
Tableau Server on Linux Administrator Guide

Synopsis

tsm data-access web-data-connectors list [options]
[global options]

Options

Global options

-h, --help

Optional.

Show the command help.

-p, --password <password>

Required, along with -u or --username if no session is active.

Specify the password for the user specified in -u or --username.

If the password includes spaces or special characters, enclose it in quotes:

--password 'my password'

-s, --server <url_to_tsm>

Optional.

Use the specified address for Tableau Services Manager. The URL must start with https, include port 8850, and use the server name not the IP address, for example https://mytableauhost:8850. If no server is specified, https://<localhost | dnsname>:8850 is assumed.

--trust-admin-controller-cert

Optional.
Use this flag to trust the self-signed certificate on the TSM controller. For more information about certificate trust and CLI connections, see Connecting TSM clients.

-u, --username <user>

Required if no session is active, along with -p or --password.

Specify a user account. If you do not include this option, the command is run using credentials you signed in with.

tsm email

Use the tsm email command to view and test your SMTP configuration.

For more information about configuring SMTP, see Configure SMTP Setup.

tsm email test-smtp-connection

Run this command to test the STMP connection. When run, TSM will attempt to establish a connection with the SMTP server that you have configured for Tableau Server. TSM will also return a connection status and the details of the SMTP configuration.

In some cases, the command will return a false-positive status. For example, if your Postfix SMTP server is set to require TLS, but Tableau Server is not configured for TLS, the connection is established and TSM will report a successful connection. However, in this scenario, Postfix actually rejects the email message after TSM has connected.

Synopsis

tsm email test-smtp-connection [global options]

Global options

-h, --help

Optional.
Show the command help.

-p, --password <password>

Required, along with -u or --username if no session is active.

Specify the password for the user specified in -u or --username.

If the password includes spaces or special characters, enclose it in quotes:

--password 'my password'

-s, --server <url_to_tsm>

Optional.

Use the specified address for Tableau Services Manager. The URL must start with https, include port 8850, and use the server name not the IP address, for example https://mytableauhost:8850. If no server is specified, https://<localhost | dnsname>:8850 is assumed.

--trust-admin-controller-cert

Optional.

Use this flag to trust the self-signed certificate on the TSM controller. For more information about certificate trust and CLI connections, see Connecting TSM clients.

-u, --username <user>

Required if no session is active, along with -p or --password.

Specify a user account. If you do not include this option, the command is run using credentials you signed in with.
You can use the `tsm initialize` command to initialize Tableau Server.

**Note:** You must apply or discard pending changes before running `tsm initialize` or the initialize will fail. Apply pending changes using the `tsm pending-changes apply` command. Discard any pending changes you do not want to apply using `tsm pending-changes discard`.

**Synopsis**

```
> tsm initialize [options] [global options]
```

**Options**

- `-r, --start-server`
  
  Optional. Leave the server running after initialization is complete.

**Global options**

- `-h, --help`
  
  Optional.

  Show the command help.

- `-p, --password <password>`
  
  Required, along with `-u` or `--username` if no session is active.

  Specify the password for the user specified in `-u` or `--username`.

  If the password includes spaces or special characters, enclose it in quotes:

  `--password 'my password'`
-s, --server <url_to_tsm>

Optional.

Use the specified address for Tableau Services Manager. The URL must start with https, include port 8850, and use the server name not the IP address, for example https://mytableauhost:8850. If no server is specified, https://<localhost | dnsname>:8850 is assumed.

--trust-admin-controller-cert

Optional.

Use this flag to trust the self-signed certificate on the TSM controller. For more information about certificate trust and CLI connections, see Connecting TSM clients.

-u, --username <user>

Required if no session is active, along with -p or --password.

Specify a user account. If you do not include this option, the command is run using credentials you signed in with.

tsm jobs

You can use the tsm jobs commands to list, reconnect to, and cancel jobs.

- cancel
- list
- reconnect

tsm jobs cancel

Cancel a job on the server. Any job can be canceled before it starts running (when queued), but only backup and cleanup jobs can be canceled when they are already running.
Synopsis

tsm jobs cancel --id <jobID> [global options]

Options

-i,--id <jobID>

Required.

Id of the job to cancel.

tsm jobs list

List asynchronous jobs on the server.

Synopsis

tsm jobs list [--status <status>] [global options]

Options

-t,--status <status>

Optional.

Filter for jobs that match the given status.

tsm jobs reconnect

Reconnect to an asynchronous job to display its progress. If no job id is specified, it reconnects to the latest job.

Synopsis

tsm jobs reconnect[--id <jobID>] [global options]
Tableau Server on Linux Administrator Guide

Options

-i, --id <jobID>

Optional.

Specifies the id of the job that should be reconnected.

Global options

-h, --help

Optional.

Show the command help.

-p, --password <password>

Required, along with -u or --username if no session is active.

Specify the password for the user specified in -u or --username.

If the password includes spaces or special characters, enclose it in quotes:

--password 'my password'

-s, --server <url_to_tsm>

Optional.

Use the specified address for Tableau Services Manager. The URL must start with https, include port 8850, and use the server name not the IP address, for example https://mytableauhost:8850. If no server is specified, https://<localhost | dnsname>:8850 is assumed.

--trust-admin-controller-cert

Optional.
Use this flag to trust the self-signed certificate on the TSM controller. For more information about certificate trust and CLI connections, see Connecting TSM clients.

-\(u\), --username <user>

Required if no session is active, along with -\(p\) or --password.

Specify a user account. If you do not include this option, the command is run using credentials you signed in with.

tsm licenses

You can use the tsm licenses commands to manage server license tasks like activating or deactivating a Tableau Server product key on- or off-line, and getting associated files for offline activation or deactivation.

- tsm licenses activate
- tsm licenses deactivate
- tsm licenses get-offline-activation-file
- tsm licenses get-offline-deactivation-file
- tsm licenses list
- tsm licenses refresh

tsm licenses activate

Activates a Tableau Server product key.

Synopsis

tsm licenses activate --license-key <product-key>
[global options]

Options

-\(f\), --license-file <file.tlf>
tsm licenses activate

Required if doing offline activation.

Specifies the license file (<file>.tlf) used for offline activation.

-k, --license-key <product-key>

Required if activating a valid product key.

Specifies the product key to use for online activation.

-t, --trial

Required if activating a trial license.

Activate a trial license.

tsrm licenses deactivate

Deactivates a Tableau Server product key either online or offline.

Synopsis

tsm licenses deactivate --license-key <product-key>
[global options]

Options

-f, --license-file <return_file.tlr>

Required if doing offline deactivation.

Specifies the license file (<file>.tlf) used for offline deactivation.

-k, --license-key <product-key>

Required if deactivating a product key.

Specifies the product key to use for online deactivation.
tsm licenses get-offline-activation-file

Generate an offline activate file to use for activating Tableau Server offline. To learn more, see Activate Tableau Server Offline - Adding a License.

Synopsis

tsm licenses get-offline-activation-file --license-key
<product-key> --output-dir <path> [global options]

Options

-k, --license-key <product-key>
   Required.
   Specifies the product key to use for offline activation.

-o, --output-dir <path>
   Required.
   The location where the offline activation file should be saved. This location must exist.

tsm licenses get-offline-deactivation-file

Generate an offline deactivation file to use for deactivating Tableau Server offline. To learn more, see Deactivate Tableau Server Offline.

Synopsis

tsm licenses get-offline-deactivation-file --license-key
<product-key> --output-dir <path> [global options]

Options

-k, --license-key <product-key>
   Required.
Specifies the product key to use for offline deactivation.

-o, --output-dir <path>

Required.

The existing location where the offline deactivation file should be saved.

tsm licenses list

List licenses on Tableau Server. The output reflects whether Tableau Server uses a user-based license, a core-based license, or a combination of the two. This command provides product keys, license expiration dates, maintenance expiration dates, and quotas for each type of user-based license. To learn more, see View Server Licenses.

Synopsis

tsm licenses list [global options]

tsm licenses refresh

Update the maintenance expiration date of all product keys on Tableau Server.

Synopsis

tsm licenses refresh [global options]

Global options

-h, --help

Optional.

Show the command help.

-p, --password <password>
Required, along with -u or --username if no session is active.

Specify the password for the user specified in -u or --username.

If the password includes spaces or special characters, enclose it in quotes:

```
--password 'my password'
```

-s, --server <url_to_tsm>

Optional.

Use the specified address for Tableau Services Manager. The URL must start with https, include port 8850, and use the server name not the IP address, for example https://mytableauhost:8850. If no server is specified, https://<localhost | dnsname>:8850 is assumed.

--trust-admin-controller-cert

Optional.

Use this flag to trust the self-signed certificate on the TSM controller. For more information about certificate trust and CLI connections, see Connecting TSM clients.

-u, --username <user>

Required if no session is active, along with -p or --password.

Specify a user account. If you do not include this option, the command is run using credentials you signed in with.

tsm login

Use the tsm login command to log in to Tableau Services Manager from a remote node.

If the account you are logged in as is a member of the TSM-authorized group, you do not need to provide credentials to run commands when running tsm CLI locally. For more information, see Authenticating with tsm CLI.
Synopsis

tsm login [global options]

Global options

-h, --help

Optional.
Show the command help.

-p, --password <password>

Required, along with -u or --username if no session is active.
Specify the password for the user specified in -u or --username.
If the password includes spaces or special characters, enclose it in quotes:
--password 'my password'

-s, --server <url_to_tsm>

Optional.
Use the specified address for Tableau Services Manager. The URL must start with https, include port 8850, and use the server name not the IP address, for example https://mytableauhost:8850. If no server is specified, https://<localhost | dnsname>:8850 is assumed.

--trust-admin-controller-cert

Optional.
Use this flag to trust the self-signed certificate on the TSM controller. For more information about certificate trust and CLI connections, see Connecting TSM clients.
-u, --username <user>

Required if no session is active, along with -p or --password.

Specify a user account. If you do not include this option, the command is run using credentials you signed in with.

**tsm logout**

You can use the `tsm logout` command to log out of Tableau Services Manager (TSM).

**Synopsis**

```
tsm logout [global options]
```

**Global options**

-**h, --help**
  
  Optional.
  
  Show the command help.

-**-p, --password <password>**
  
  Required, along with -u or --username if no session is active.
  
  Specify the password for the user specified in -u or --username.

  If the password includes spaces or special characters, enclose it in quotes:

  ```
  --password 'my password'
  ```

-**-s, --server <url_to_tsm>**
  
  Optional.
Use the specified address for Tableau Services Manager. The URL must start with https, include port 8850, and use the server name not the IP address, for example https://mytableauhost:8850. If no server is specified, https://<localhost | dnsname>:8850 is assumed.

--trust-admin-controller-cert

Optional.

Use this flag to trust the self-signed certificate on the TSM controller. For more information about certificate trust and CLI connections, see Connecting TSM clients.

-u, --username <user>

Required if no session is active, along with -p or --password.

Specify a user account. If you do not include this option, the command is run using credentials you signed in with.

tsrm maintenance

You can use the tsm maintenance commands to manage server maintenance tasks like creating regular backups or restoring Tableau Server from a previously created backup.

- tsm maintenance backup
- tsm maintenance cleanup
- tsm maintenance metadata-services
  - tsm maintenance metadata-services enable
  - tsm maintenance metadata-services disable
  - tsm maintenance metadata-services get-status
- tsm maintenance reindex-search
- tsm maintenance reset-searchserver
- tsm maintenance restore
tsm maintenance backup

Creates a backup of the data managed by Tableau Server. This data includes the Tableau PostgreSQL database (the repository) which contains workbook and user metadata, and extract (.tde or .hyper) files. This data does not include configuration data. See Perform a Full Backup and Restore of Tableau Server.

Locating the repository (pgsql) and the File Store on the same node as the Administration Controller can reduce the length of time it takes to back up Tableau Server. The Administration Controller is usually on the initial node, unless you have had an initial node failure and had to move the controller to another node.

When the repository is co-located with the controller it eliminates or greatly reduces the need to compress and transfer data between nodes during the backup process. This is also true with the File Store, especially if your organization uses extracts heavily.

Synopsis

```
  tsm maintenance backup --file <backup_file> [options]
  [global options]
```

Options

- `--append-date`

Optional.

Append the current date to the backup file name.
Tableau Server on Linux Administrator Guide

-f, --file <backup_file>

Required.

Write the backup to the specified file. An extension of .tsbak is automatically used. The file is written to the directory defined in the TSM basefilepath.backuprestore variable. By default:

/var/opt/tableau/tableau_server/data/tabsvc/files/backups/

For more information about file paths and how to change them, see tsm File Paths.

-i, --description <string>

Optional.

Include the specified description of the backup file.

--skip-compression

Optional.

Create a backup without using compression. This results in a larger backup file but can reduce the amount of time it takes to complete the backup. If using this in a multi-node installation, we strongly recommend you have a File Store instance configured on your initial node.

-k, --skip-verification

Optional.

Do not verify the integrity of the database backup.

--override-disk-space-check

Optional.
Attempt to create a backup even when there is a low disk space warning.

-po, --pg-only

Optional.

Generates only the repository backup.

--request-timeout <timeout in seconds>

Optional.

Wait the specified amount of time for the command to finish. Default value is 1800 (30 minutes).

Examples

This example creates a backup called ts_backup-<yyyy-mm-dd>.tsbak in the
/var/opt/tableau/tableau_server/data/tabsvc/files/backups/ directory:

tsm maintenance backup -f ts_backup -d

tsm maintenance cleanup

Deletes old log files and temporary files.

If you are running Tableau Server on a distributed deployment, run this command on the node that is running the Administration Controller (also referred to as the TSM Controller) process. By default, the controller is on the initial node in the cluster.

The purpose of this command is to create free space on the Tableau deployment by deleting old files. Therefore, running this command when the server is started will remove the majority of old content for this scenario. To remove current log files, then you must stop the server before running this command.
This command was added in Tableau Server version 10.5.1 and some options were added in version 2018.1.

Synopsis

tsm maintenance cleanup [options] [global options]

Options

-a, --all

Optional.

Perform all cleanup operations with default retention values. Equivalent to running the cleanup command with the following options: -l -t -r -q.

--http-requests-table-retention <# of days>

Optional.

Default: 7 days

Specify the number of days of http_requests table entries that should be retained. Entries older than the specified number of days are deleted. This option specifies the retention age but does not delete table entries. To delete entries, use the -q option in addition to this option.

-ic, --sheet-image-cache

Optional.

Clear the image cache. This cache can contain images for offline previews, snapshots for subscription email messages, and subscription pdfs, as well as any images requested from the publish rest API endpoint (see rest_api_ref.htm for more information).

Note: Option added in version 2019.4
-1, --log-files

Optional.

Delete log files that are older than the retention-period. Files in the subdirectories under data/tabsvc/logs will be deleted.

--log-files-retention <# of days>

Optional.

Default: 1 (24 hours)

Delete logs older than this number of days. This command does not apply to temporary files.

-q, --http-requests-table

Optional.

Delete old http_requests table entries. Tableau Server must be running for table entries to be deleted. This option is ignored if Tableau Server is stopped.

-r, --redis-cache

Optional.

Clear the Redis cache.

--request-timeout <timeout in seconds>

Optional.

Wait the specified amount of time for the command to finish. Default value is 1800 (30 minutes).

-t, --temp-files

Optional.
Delete all files and subdirectories in the following directories:

- `/var/opt/tableau/tableau_server/data/tabsvc/temp`: Only directories that are storing files for expired (not running) sessions are deleted.
- `/var/opt/tableau/tableau_server/data/tabsvc/httpd/temp`
- `/var/opt/tableau/tableau_server/temp`

```
-v, --verbose
```

Optional.

Show details of what was cleaned.

Examples

This example cleans up all log files older than 2 days old:

```
tsm maintenance cleanup -l --log-files-retention 2
```

```
tsm maintenance metadata-services enable
```

Use the `tsm maintenance metadata-services enable` command to enable the Tableau Metadata API for Tableau Server.

If Tableau Server is licensed with the Data Management Add-on, enabling the Metadata API enables Tableau Catalog.

When enabling the Metadata API, information about the content on Tableau Server is ingested and then indexed to the Metadata API Store. The Metadata API can be used to query schema, lineage, and user managed metadata about the content published to Tableau Server. After the Metadata API is enabled, metadata is continuously ingested and indexed until the Metadata API is disabled.

When running this command, keep the following in mind:

- This command stops and starts some services used by Tableau Server, which causes certain functionality, such as Recommendations, to be temporarily unavailable to your
users.

- A new index of metadata is created and replaces the previous index every time this command is used.

For more information about the Tableau Catalog, see, About Tableau Catalog.

Synopsis

tsm maintenance metadata-services enable

Option

--ignore-prompt

Optional.

Dismiss the confirmation prompt when enabling the Metadata API.

tsm maintenance metadata-services disable

Use the tsm maintenance metadata-services disable command to disable the Tableau Metadata API.

Disabling the Metadata API stops continuous ingestion and indexing of information about the content on Tableau Server, deletes the index of information about the content published to Tableau Server and assets associated with that content, and disables the ability to both query the Metadata API and access Tableau Catalog.

Running this command stops and starts some services used by Tableau Server, which causes certain functionality, such as Recommendations, to be temporarily unavailable to your users.

Synopsis

tsm maintenance metadata-services disable

Option

--ignore-prompt
Optional.

Dismiss the confirmation prompt when disabling the Metadata API.

tsm maintenance metadata-services get-status

Use the tsm maintenance metadata-services get-status command to get status information on Metadata Services.

Status on Metadata Services indicates if the Metadata API Store has been initialized or if the Tableau Metadata API is running or not.

Synopsis

tsm maintenance metadata-services get-status

tsm maintenance reindex-search

Use the tsm maintenance reindex-search command to rebuild the search index.

Synopsis

tsm maintenance reindex-search [options] [global options]

Option

--request-timeout <timeout in seconds>

Optional.

Wait the specified amount of time for the command to finish. Default value is 1800 (30 minutes).

tsm maintenance reset-searchserver

Resets the search server to a clean state, deleting search information and rebuilding the search index.
Synopsis

```
tsm maintenance reset-searchserver [options] [global options]
```

Option

```
--request-timeout <timeout in seconds>
```

Optional.

Wait the specified amount of time for the command to finish. Default value is 1800 (30 minutes).

**tsm maintenance restore**

Restore Tableau Server using the specified backup file. Restoring a backup file does not restore any configuration data. See Perform a Full Backup and Restore of Tableau Server.

You can only restore from a backup that has the same type of identity store as the running server. For example, a backup from a server using local authentication can be restored to a Tableau Server initialized with local authentication, but a backup from a server using Active Directory authentication cannot be restored to a server initialized with local authentication.

Synopsis

```
tsm maintenance restore --file <file_name> [--restart-server] [global options]
```

Options

```
-ak, --asset-key-file <file_name>
```

Optional. Specify this option only if you are restoring from assets that were created by tabadmin on Tableau Server (versions 2018.1 and earlier).
Name of asset key file to restore from. The asset key file is created by the `tabadmin assetkeys` command. The file must be in the predefined backup/restore location on the server.

`-f`, `--file <file_name>`

Required.

Specifies the backup file to restore from.

The `restore` command expects a backup file in the directory defined in the TSM `basefilepath.backuprestore` variable. By default:

```
/var/opt/tableau/tableau_server/data/tabsvc/files/backups/
```

For more information about file paths and how to change them, see tsm File Paths.

`-k`, `--skip-identity-store-verification`

Optional. Specify this option only if you are restoring from a backup file that was created by `tabadmin` on Tableau Server (versions 2018.1 and earlier).

Skip identity store verification. Specify this option if you are restoring a backup file that was created with the `tabadmin backup --no-config` command.

`-po`, `--pg-only`

Optional.

Restores only the repository.

`-r`, `--restart-server`

Optional.

Restart the server after the restore.

`--request-timeout <timeout in seconds>`
Optional.

Wait the specified amount of time for the command to finish. Default value is 1800 (30 minutes).

tsm maintenance send-logs

Upload the specified file to Tableau and associate it with a support case. To successfully upload files to Tableau, your Tableau Server must be able to communicate with the send-logs server at https://report-issue.tableau.com.

Synopsis

tsm maintenance send-logs --case <case_number> --email <contact_email> --file <path/to/file> [global options]

Options

-c,--case <case_number>

Required.

Support case number.

-e,--email <contact_email>

Required.

Contact email.

-f, --file <path/to/file>

Required.

Specifies the location and name of the log file archive to send.

--request-timeout <timeout in seconds>

Optional.
Wait the specified amount of time for the command to finish. Default value is 1800 (30 minutes).

**tsm maintenance validate-resources**

Validate workbooks and data sources for a site.

**Synopsis**

```bash
tsm maintenance validate-resources --site-id <site ID>
[global options]
```

**Options**

- `-id, --site-id <site ID>`
  - Required.
  - ID for the site whose resources you are validating.

- `-r, --repair`
  - Optional.
  - Repair invalid resources.

- `--request-timeout <timeout in seconds>`
  - Optional.
  - Wait the specified amount of time for the command to finish. Default value is 1800 (30 minutes).

**tsm maintenance ziplogs**

Use the `ziplogs` command to create an archive of Tableau Server log files.
Note: If you cannot run the ziplogs command successfully, you can manually zip the Tableau Server logs. For more information, see Troubleshoot Tableau Server on Linux.

Synopsis

tsm maintenance ziplogs [options] [global options]

Options

-a, --all

Optional.

Include all files except PostgreSQL data.

-d, --with-postgresql-data

Optional.

Include the PostgreSQL data folder if Tableau Server is stopped or PostgreSQL dump files if Tableau Server is running.

--enddate <mm/dd/yyyy>

Optional.

The last date of log files to be included. This option must be used with --startdate and cannot be used with --minimumdate. If this option is not specified, up to two days of logs will be included.

Added in version 2019.3

-f, --file <name>

Optional.
Specify a name for the zipped archive file. If no name is provided the archive is created as logs.zip. The file is written to the directory defined in the TSM basefilepath.log_archive variable. By default:

/var/opt/tableau/tableau_server/data/tabsvc/files/log-archives/

For more information about file paths and how to change them, see tsm File Paths.

-i, --description <string>

Optional.

Include the specified description of the archive file.

-l, --with-latest-dump

Optional.

When any service crashes, Tableau Server generates a dumpfile. Set this option to include the most recent service crash dumpfile. If you do not set this option, then no dumpfile will be included in the resulting ziplog.

-m, --minimumdate <mm/dd/yyyy>

Optional.

Earliest date of log files to be included. If not specified, a maximum of two days of log files are included. Format of date should be "mm/dd/yyyy". This option cannot be used with --startdate and --enddate.

-o, --overwrite

Optional.

For an overwrite of an existing ziplog file. If a file by the same name already exists and this option is not used, the ziplogs command will fail.

By default the file is written to:
/var/opt/tableau/tableau_server/data/tabsvc/files/log-archives/  

For more information about file paths and how to change them, see tsm File Paths.

--startdate <mm/dd/yyyy>

Optional.

The earliest date of log files to be included. This option must be used with --enddate and cannot be used with --minimumdate. If this option is not specified, up to two days of logs will be included.

Added in version 2019.3

--request-timeout <timeout in seconds>

Optional.

Wait the specified amount of time for the command to finish. Default value is 7200 (120 minutes).

-t, --with-netstat-info

Optional.

Include netstat information.

Global options

-h, --help

Optional.

Show the command help.

-p, --password <password>
Required, along with -u or --username if no session is active.

Specify the password for the user specified in -u or --username.

If the password includes spaces or special characters, enclose it in quotes:

--password 'my password'

-s, --server <url_to_tsm>

Optional.

Use the specified address for Tableau Services Manager. The URL must start with https, include port 8850, and use the server name not the IP address, for example https://mytableauhost:8850. If no server is specified, https://<localhost | dnsname>:8850 is assumed.

--trust-admin-controller-cert

Optional.

Use this flag to trust the self-signed certificate on the TSM controller. For more information about certificate trust and CLI connections, see Connecting TSM clients.

-u, --username <user>

Required if no session is active, along with -p or --password.

Specify a user account. If you do not include this option, the command is run using credentials you signed in with.

tsm pending-changes

You can use the tsm pending-changes commands to apply, discard, or view pending configuration and topology changes to Tableau Server.
Passwords and secrets that you enter during TSM configuration are encrypted after you save them. Secrets remain encrypted until, during, and after you apply pending changes. For more information about secret storage, see Manage Server Secrets.

- tsm pending-changes apply
- tsm pending-changes discard
- tsm pending-changes list

**tsm pending-changes apply**

You can use the `tsm pending-changes apply` command to apply pending configuration and topology changes to Tableau Server.

A prompt warns you that the command will restart Tableau Server. If Tableau Server is currently running it is restarted, and if it is stopped it is left in a stopped state with no restart. You can suppress the prompt using the `--ignore-prompt` option, but this does not change the restart behavior.

**Synopsis**

`tsm pending-changes apply [global options]`

**Options**

- `-iw, --ignore-warnings`
  
  Optional.
  
  Ignore warning level constraints.

- `--ignore-prompt`
  
  Optional.
Suppress the prompt for restart. This only suppresses the prompt. The restart behavior is unchanged.

--request-timeout <timeout in seconds>

Optional.

Wait the specified amount of time for the command to finish. Default value is 1800 (30 minutes).

tsm pending-changes discard

You can use the tsm pending-changes discard command to discard pending configuration and topology changes to Tableau Server.

Synopsis

tsm pending-changes discard [options] [global options]

Options

--config-only

Optional.

Discard only pending configuration changes.

--topology-only

Optional.

Discard only pending topology changes.

tsm pending-changes list

You can list pending configuration and topology changes to Tableau Server.

Synopsis

tsm pending-changes list [options] [global options]
Options

--config-only

Optional.
List only pending configuration changes.

--topology-only

Optional.
List only pending topology changes.

Global options

-h, --help

Optional.
Show the command help.

-p, --password <password>

Required, along with -u or --username if no session is active.
Specify the password for the user specified in -u or --username.
If the password includes spaces or special characters, enclose it in quotes:

--password 'my password'

-s, --server <url_to_tsm>

Optional.
Use the specified address for Tableau Services Manager. The URL must start with https, include port 8850, and use the server name not the IP address, for example
--trust-admin-controller-cert

Optional.

Use this flag to trust the self-signed certificate on the TSM controller. For more information about certificate trust and CLI connections, see Connecting TSM clients.

-u, --username <user>

Required if no session is active, along with -p or --password.

Specify a user account. If you do not include this option, the command is run using credentials you signed in with.

tsm register

You can use the `tsm register` command to register Tableau Server. This command lets you either display the registration file template (using the `--template` option) or provide the path to a completed registration file (using the `--file` option). You must use one of these two options when calling the `tsm register` command.

Synopsis

`tsm register --template | --file <registration-filename>`

[global options]

Options

--file <registration-filename>

Required.

Path to the file that contains the registration data.
--template

Required.
Display registration file template.

Global options

-h, --help

Optional.
Show the command help.

-p, --password <password>

Required, along with -u or --username if no session is active.
Specify the password for the user specified in -u or --username.
If the password includes spaces or special characters, enclose it in quotes:
--password 'my password'

-s, --server <url_to_tsm>

Optional.
Use the specified address for Tableau Services Manager. The URL must start with https, include port 8850, and use the server name not the IP address, for example https://mytableauhost:8850. If no server is specified, https://<localhost | dnsname>:8850 is assumed.

--trust-admin-controller-cert

Optional.
Use this flag to trust the self-signed certificate on the TSM controller. For more information about certificate trust and CLI connections, see Connecting TSM clients.

-u, --username <user>

Required if no session is active, along with -p or --password.

Specify a user account. If you do not include this option, the command is run using credentials you signed in with.

tsm reset

Use the tsm reset command to clear the initial admin user so that you can enter a new one. After you run tsm reset you must rerun the tabcmd initialuser command to create a new initial admin. The new name cannot be the same username as the previous admin user.

If your organization is using Active Directory or LDAP for the Tableau identity store, then the account and password you specify must match an account in the directory.

Synopsis

tsm reset[option] [global options]

Option

-d,--delete-all-sessions

Optional.

Delete all active user sessions when the server is reset.

--request-timeout <timeout in seconds>

Optional.
Wait the specified amount of time for the command to finish. Default value is 1800 (30 minutes).

Global options

-h, --help
Optional.
Show the command help.

-p, --password <password>
Required, along with -u or --username if no session is active.
Specify the password for the user specified in -u or --username.
If the password includes spaces or special characters, enclose it in quotes:
--password 'my password'

-s, --server <url_to_tsm>
Optional.
Use the specified address for Tableau Services Manager. The URL must start with https, include port 8850, and use the server name not the IP address, for example https://mytableauhost:8850. If no server is specified, https://<localhost | dnsname>:8850 is assumed.

--trust-admin-controller-cert
Optional.
Use this flag to trust the self-signed certificate on the TSM controller. For more information about certificate trust and CLI connections, see Connecting TSM clients.

-u, --username <user>
Required if no session is active, along with -p or --password.

Specify a user account. If you do not include this option, the command is run using credentials you signed in with.

**tsm restart**

You can use the *tsm restart* command to restart Tableau Server.

**Synopsis**

```
  tsm restart [global options]
```

**Option**

```
--request-timeout <timeout in seconds>
```

Required, along with -u or --username if no session is active.

Specify a user account. If you do not include this option, the command is run using credentials you signed in with.

```
  --password <password>
```

Optional.

Show the command help.

Required, along with -u or --username if no session is active.

Specify the password for the user specified in -u or --username.

If the password includes spaces or special characters, enclose it in quotes:

```
  --password 'my password'
```
-s, --server <url_to_tsm>

Optional.

Use the specified address for Tableau Services Manager. The URL must start with https, include port 8850, and use the server name not the IP address, for example https://mytableauhost:8850. If no server is specified, https://<localhost | dnsname>:8850 is assumed.

--trust-admin-controller-cert

Optional.

Use this flag to trust the self-signed certificate on the TSM controller. For more information about certificate trust and CLI connections, see Connecting TSM clients.

-u, --username <user>

Required if no session is active, along with -p or --password.

Specify a user account. If you do not include this option, the command is run using credentials you signed in with.

tsm security

Use the tsm security commands to configure Tableau Server support for external (gateway) SSL or repository (Postgres) SSL. Repository SSL configuration includes the option to enable SSL over direct connections from Tableau clients—including Tableau Desktop, Tableau Mobile, and web browsers—to the repository.

- tsm security custom-cert
  - tsm security custom-cert add
  - tsm security custom-cert delete
  - tsm security custom-cert list
Prerequisites

Before you configure SSL, you must acquire certificates, and then copy them to the computer that runs the Tableau Server gateway process. Additional preparation is required for enabling direct connections from clients. To learn more, see the following articles:

Configure SSL for External HTTP Traffic to and from Tableau Server

Configure SSL for Internal Postgres Communication

For information about mutual (two-way) SSL, see Configure Mutual SSL Authentication and tsm authentication mutual-ssl commands.

tsm security custom-cert add

Adds a custom CA certificate to Tableau Server. This certificate is optionally used to establish trust for TLS communication between a SMTP server and Tableau Server.

If a custom certificate already exists, this command will fail. You can remove the existing custom certificate using the tsm security custom-cert delete command.
Note: The certificate that you add with this command may be used by other Tableau Server services for TLS connections.

Synopsis

tsm security custom-cert add --cert-file <file.crt>
[global options]

Options

-c, --cert-file <file.crt>

Required. Specify the name of a certificate file in valid PEM or DER format.

tsm security custom-cert delete

Removes the server’s existing custom certificate. Doing this allows you to add a new custom certificate.

Synopsis

tsm security custom-cert delete[global options]

 tsm security custom-cert list

List details of custom certificate.

Synopsis

tsm security custom-cert list[global options]

 tsm security external-ssl disable

Removes the server’s existing SSL configuration settings and stops encrypting traffic between external clients and the server.
Synopsis

tsm security external-ssl disable [global options]

tsm security external-ssl enable

Enable and specify certificate and key files for SSL over external HTTP communication.

Synopsis

tsm security external-ssl enable --cert-file <file.crt> --key-file <file.key> [options] [global options]

Options

--cert-file <file.crt>

Required. Specify the name of a valid PEM-encoded x509 certificate with the extension .crt.

--key-file <file.key>

Required. Specify a valid RSA or DSA private key file, with the extension .key by convention.

--chain-file <chainfile.crt>

Specify the certificate chain file (.crt)

A certificate chain file is required for Tableau Desktop on the Mac. In some cases, a certificate chain file may be required for Tableau Mobile.

Some certificate providers issue two certificates for Apache. The second certificate is a chain file, which is a concatenation of all the certificates that form the certificate chain for the server certificate.

All certificates in the file must be x509 PEM-encoded and the file must have a .crt extension (not .pem).
--passphrase

Optional. Passphrase for the certificate file. The passphrase you enter will be encrypted while at rest.

Note: If you create a certificate key file with a passphrase, you cannot reuse the SSL certificate key for SAML.

--protocols <list protocols>

Optional. List the Transport Layer Security (TLS) protocol versions you want to allow or disallow.

TLS is an improved version of SSL. Tableau Server uses TLS to authenticate and encrypt connections. Accepted values include protocol versions supported by Apache. To disallow a protocol, prepend the protocol version with a minus (-) character.

Default setting: "all, -SSLv2, -SSLv3"

This default explicitly does not allow clients to use SSL v2 or SSL v3 protocols to connect to Tableau Server. However, we recommend that you also disallow TLS v1 and TLS v1.1.

Before you deny a specific version of TLS, verify that the browsers from which your users connect to Tableau Server support TLS v1.2. You might need to preserve support for TLSv1.1 until browsers are updated.

If you do not need to support TLS v1 or v1.1, use the following command to allow TLS v1.2 (using the value all), and explicitly deny SSL v2, SSL v3, TLS v1, and TLS v1.1.

```bash
tsm security external-ssl enable --cert-file file.crt --key-file file.key --protocols "all -SSLv2 -SSLv3 -TLSv1 -TLSv1.1"
```

tsm security external-ssl list

Displays a list of settings related to the configuration of gateway external SSL. The list includes the names of the certificate files in use, but not their location.
Synopsis

```
  tsm security external-ssl list [global options]

  tsm security regenerate-internal-tokens
```

This command performs the following operations:

1. Generates new internal SSL certificates for Postgres repository the search server.
2. Generates new passwords for all of the internally managed passwords.
3. Updates all Postgres repository passwords.
4. Generates a new encryption key for asset key management and encrypts the asset key data with the new key.
5. Generates a new encryption key for configuration secrets (master key) and encrypts the configuration with it.
6. Reconfigures and updates Tableau Server with all of these secrets. In a distributed deployment, this command also distributes the reconfiguration and updates across all nodes in the cluster.
7. Stops the server.
8. Regenerates a new master key, adds it to the master keystore file, and then creates new security tokens for internal use.
9. Starts the server.

If you plan to add a node to your cluster after you have run this command, then you will need to generate a new node configuration file to update the tokens, keys, and secrets that are generated by this command. See Install and Configure Additional Nodes.
Synopsis

tsm security regenerate-internal-tokens [options]
[global options]

Options

--request-timeout <timeout in seconds>

  Optional.
  
  Wait the specified amount of time for the command to finish. Default value is 1800 (30 minutes).

tsm security repository-ssl disable

Stop encrypting traffic between the repository and other server components, and stop support for direct connections from Tableau clients.

Synopsis

tsm security repository-ssl disable [global-options]

tsm security repository-ssl enable

Enables SSL and generates the server’s .crt and .key files used for encrypted traffic between the Postgres repository and other server components. Enabling this also gives you the option to enable SSL over direct connections from Tableau clients to the server.

Synopsis

tsm security repository-ssl enable [options] [global options]

Options

-i, --internal-only

  Optional. When set to --internal-only, Tableau Server uses SSL between the
repository and other server components, and it supports but does not require SSL for direct connections through `tableau` or `readonly` users.

If this option is not set, Tableau Server requires SSL for traffic between the repository and other server components, as well as for direct connections from Tableau clients (for connections through the `tableau` or `readonly` users).

When you specify this option, you must also complete the steps described in Configure Postgres SSL to Allow Direct Connections from Clients.

tsm security repository-ssl get-certificate-file

Get the public certificate file used for SSL communication with the Tableau repository. SSL must be enabled for repository communication before you can retrieve a certificate. The certificate file is distributed automatically to internal clients of the repository in the Tableau Server cluster. To enable remote clients to connect over SSL to the repository, you must copy the public certificate file to each client.

Synopsis

tsm security repository-ssl get-certificate-file [global-options]

Options

- `-f, --file`

  Required.

  Full path and file name (with .cert extension) where the certificate file should be saved. If a duplicate file exists it will be overwritten.

tsm security repository-ssl list

Returns the existing repository (Postgres) SSL configuration.

Synopsis

tsm security repository-ssl list [global-options]
tsm security vizql-extsvc-ssl disable

Disables the connection to external service connection for SCRIPT functions.

Synopsis

tsm security vizql-extsvc-ssl disable [global options]

tsm security vizql-extsvc-ssl enable

Enables and configures connection to external service connection for SCRIPT functions.
Tableau Server supports SSL for Rserve or TabPy connections.

If you are updating or changing an existing external services configuration, run the tsm security vizql-extsvc-ssl disable command to clear the settings before you run this command.

Synopsis

tsm security vizql-extsvc-ssl enable --connection-type <type> --extsvc-host <host_name> --extsvc-port <port> [options] [global options]

Options

--connection-type <type>

Specify the external service type that you are configuring. Valid values are: rserve-secure, rserve, tabpy-secure, tabpy.

--extsvc-host <host_name>

Required. Specify the host name or IP address of the server in your organization that is hosting the external service.

--extsvc-port <port_number>

Required. Specify the port that is used to connect to the external service. Default
value for Rserve is typically \texttt{6311}. Default value for TabPy is typically \texttt{9004}.

- \texttt{cf, --cert-file <file.crt>}

  Optional. Specify the path and filename of a valid PEM-encoded x509 certificate with the extension .crt.

- \texttt{--extsvc-username <user_name>}

  Optional. If the connection to the external service requires authentication, specify the user name.

- \texttt{--extsvc-password <password>}

  Optional. If the connection to the external service requires authentication, specify the password.

- \texttt{--connect-timeout-ms <milliseconds>}

  Optional. Connection timeout in milliseconds. Default is \texttt{1000}. Raise the value of this setting if Tableau is timing out before the external server can respond.

- \texttt{--script-disabled <true | false>}

  Optional. Disable scripts originating from the external service from running on Tableau Server. Default is \texttt{true}. To allow scripts from the external service to run on Tableau Server, set to \texttt{false}.

\texttt{tsm security vizql-extsvc-ssl list}

Displays a list of settings related to the configuration of external service SSL connection for SCRIPT functions. The list includes the names of the certificate files in use, host name, port, user name, timeout duration, and other details.

\textbf{Synopsis}

\texttt{tsm security vizql-extsvc-ssl list [global options]}
Global options

-h, --help

Optional.
Show the command help.

-p, --password <password>

Required, along with -u or --username if no session is active.
Specify the password for the user specified in -u or --username.
If the password includes spaces or special characters, enclose it in quotes:
--password 'my password'

-s, --server <url_to_tsm>

Optional.
Use the specified address for Tableau Services Manager. The URL must start with https, include port 8850, and use the server name not the IP address, for example https://mytableauhost:8850. If no server is specified, https://<localhost | dnsname>:8850 is assumed.

--trust-admin-controller-cert

Optional.
Use this flag to trust the self-signed certificate on the TSM controller. For more information about certificate trust and CLI connections, see Connecting TSM clients.

-u, --username <user>

Required if no session is active, along with -p or --password.
Specify a user account. If you do not include this option, the command is run using credentials you signed in with.

**tsm settings**

You can use the `tsm settings` commands to export (get) and import (set) configuration values.

- `tsm settings export`
- `tsm settings import`

**Important:** The server configuration file referenced in this topic includes a copy of the master keystore file used for encrypting configuration secrets. We strongly recommend that you take additional measures to secure the node configuration file, using mechanisms as described in Securing secrets for import and export operations.

**tsm settings export**

Export the current server configuration and topology to a file.

**Synopsis**

```
  tsm settings export --output-config-file <path/to/output_file.json> [global options]
```

**Options**

- `-f, --output-config-file <file>`
  
  Required.

  Specifies the location and name of the file created by this operation.
tsm settings import

Import server configuration or topology.

Synopsis

```
  tsm settings import --import-config-file <path/to/import_file.json> [global options]
```

Options

```
  -f, --import-config-file <FILE>
```

Required.

Path to input file.

```
  --config-only
```

Optional.

```
  --topology-only
```

Optional.

```
  -frc, --force-keys
```

Optional.

Force a key to be added to configuration even if it did not previously exist.

Global options

```
  -h, --help
```

Optional.
Show the command help.

-p, --password <password>

Required, along with -u or --username if no session is active.

Specify the password for the user specified in -u or --username.

If the password includes spaces or special characters, enclose it in quotes:

--password 'my password'

-s, --server <url_to_tsm>

Optional.

Use the specified address for Tableau Services Manager. The URL must start with https, include port 8850, and use the server name not the IP address, for example https://mytableauhost:8850. If no server is specified, https://<localhost | dnsname>:8850 is assumed.

--trust-admin-controller-cert

Optional.

Use this flag to trust the self-signed certificate on the TSM controller. For more information about certificate trust and CLI connections, see Connecting TSM clients.

-u, --username <user>

Required if no session is active, along with -p or --password.

Specify a user account. If you do not include this option, the command is run using credentials you signed in with.
tsm sites

You can use the `tsm sites` commands to export an existing site for import to a new site (also referred to as site migration), and to import the new site. An `unlock` command is available in case an error leaves a site locked.

The `tsm sites` commands will use your local file store to hold the export and import data. If you are running a multinode Tableau cluster, then you must run the `tsm sites` commands on a Tableau Server that is running the Data Engine process. For information about the Data Engine process and the processes that require it, see Tableau Server Processes.

For comprehensive steps for migrating a site, see Export or Import a Site.

- tsm sites export
- tsm sites import
- tsm sites import-verified
- tsm sites unlock

tsm sites export

Export a specified Tableau Server site to a .zip file. You can export a site to archive its settings at a specific point in time, or to complete the first step of a site migration process.

**Note:** The `tsm sites import` and `tsm sites export` commands can leave a site in a locked state if an error occurs. To unlock a site, use the `tsm sites unlock` command.

**Synopsis**

```
tsm sites export --site-id <source-siteID> --file <export-file> [options] [global options]
```
Options

-f, --file <export-file>

Required.

Specify the name of the file to which Tableau Server saves all of the site's information.

This file is generated to the directory defined in the TSM basefilepath.site_export.exports variable. By default:

/var/opt/tableau/tableau_server/data/tabsvc/files/siteexports

For more information about file paths and how to change them, see tsm File Paths.

-id, --site-id <source-siteID>

Required.

The site ID for the site you are exporting. You can get the site ID from the URL when you're signed in to the site from a web browser. For information about locating the site ID, see Prepare the source and target sites.

-ow, --overwrite

Optional.

Overwrite an export file of the same name that already exists.

--request-timeout <timeout in seconds>

Optional.

Wait the specified amount of time for the command to finish. Default value is 1800 (30 minutes).
tsm sites import

This command uses the .zip file you created using `tsm sites export` to generate a set of .csv files that show how the exported source site settings will map to the new target site.

By default, the .zip file is generated and saved to the `siteexports` directory at:

```
/var/opt/tableau/tableau_server/data/tabsvc/files/siteexports
```

Before you use this command, you must copy the .zip file to the directory in which Tableau will expect it. This location is defined in the TSM `basefilepath.site_import.-exports` variable. By default, the import directory is:

```
/var/opt/tableau/tableau_server/data/tabsvc/files/siteimports
```

For more information about file paths and how to change them, see `tsm File Paths`.

**Note:** The `tsm sites import` and `tsm sites export` commands can leave a site in a locked state if an error occurs. To unlock a site, use the `tsm sites unlock` command.

**Synopsis**

```
tsm sites import --file <export-file.zip> --site-id <target-siteID> [options] [global options]
```

**Options**

```
-f,--file <export-file.zip>
```

Required.

Name of the .zip file created by the `tsm sites export` process, and which you must copy to the import directory. By default:

```
/var/opt/tableau/tableau_server/data/tabsvc/files/siteimports
```
Tableau Server on Linux Administrator Guide

- \texttt{id}, \texttt{--site-id} <\texttt{target-siteID}>

  Required.

  The site ID for the new site you are importing to (the target site). For information about locating the site ID, see Prepare the source and target sites.

- \texttt{c}, \texttt{--continue-on-ignorable-errors}

  Optional.

  Continue site import if errors occur which can be ignored. These errors can indicate issues with the import of a specific workbook or data source.

- \texttt{k}, \texttt{--no-verify}

  Optional.

  Skip verification of mapping files.

- \texttt{m}, \texttt{--override-schedule-mapper} <\texttt{mapping-file.csv}>

  Optional.

  Schedule mapping file to override the normal mapping by name.

\texttt{--request-timeout} <\texttt{timeout in seconds}>

  Optional.

  Wait the specified amount of time for the command to finish. Default value is 1800 (30 minutes).

**tsm sites import-verified**

Specify the directory that contains an exported site’s .csv mapping files, to import to a new site. This is the final step of a site migration process.
Synopsis

```bash
tsm sites import-verified --import-job-dir <importjob-directory> --site-id <target-siteID> [options] [global options]
```

Options

- **-id, --site-id <target-siteID>**
  
  Required.
  
  The site ID for the new site you are importing to (the target site). For information about locating the site ID, see Prepare the source and target sites.

- **-w, --import-job-dir <importjob-directory>**
  
  Required.
  
  The parent of the mappings directory that contains the .csv files from the exported (source) site. The name of this parent directory includes the import id and date and time. For example:

  ```bash
  /var/opt/tableau/tableau_server-/data/tabsvc/files/siteimports/working/import_ff00_20180102022014457
  ```

- **--request-timeout <timeout in seconds>**
  
  Optional.
  
  Wait the specified amount of time for the command to finish. Default value is 1800 (30 minutes).

**tsm sites unlock**

Use this command to unlock a site.
Tableau Server on Linux Administrator Guide

Options

-\( \text{-id}, --\text{site-id} \ <\text{target-siteID}> \)

Required.

The site ID for the site you are unlocking. For information about locating the site ID, see Prepare the source and target sites.

-\( \text{-d, --desired-state} \ <\text{state to leave unlocked site in}> \)

Optional.

The state the site should be left in after it is unlocked. Options are "active" and "suspended". The default is "active" if not specified.

For example:

\[
\text{tsm sites unlock -id mysite -d suspended}
\]

-\( \text{--request-timeout} \ <\text{timeout in seconds}> \)

Optional.

Wait the specified amount of time for the command to finish. Default value is 1800 (30 minutes).

Global options

-\( \text{-h, --help} \)

Optional.

Show the command help.

-\( \text{-p, --password} \ <\text{password}> \)

Required, along with \( \text{-u or --username} \) if no session is active.
Specify the password for the user specified in \(-u\) or \(--username\).

If the password includes spaces or special characters, enclose it in quotes:

\(--password \'my password\'

\(-s, --server <url_to_tsm>

Optional.

Use the specified address for Tableau Services Manager. The URL must start with https, include port 8850, and use the server name not the IP address, for example https://mytableauhost:8850. If no server is specified, https://<localhost | dnsname>:8850 is assumed.

\(--trust-admin-controller-cert

Optional.

Use this flag to trust the self-signed certificate on the TSM controller. For more information about certificate trust and CLI connections, see Connecting TSM clients.

\(-u, --username <user>

Required if no session is active, along with \(-p\) or \(--password\).

Specify a user account. If you do not include this option, the command is run using credentials you signed in with.

tsm start

You can use the tsm start command to start Tableau Server.

Synopsis

tsm start [option][global options]
Option

--request-timeout <timeout in seconds>

Optional.

Wait the specified amount of time for the command to finish. Default value is 1800 (30 minutes).

Global options

-h, --help

Optional.

Show the command help.

-p, --password <password>

Required, along with -u or --username if no session is active.

Specify the password for the user specified in -u or --username.

If the password includes spaces or special characters, enclose it in quotes:

--password 'my password'

-s, --server <url_to_tsm>

Optional.

Use the specified address for Tableau Services Manager. The URL must start with https, include port 8850, and use the server name not the IP address, for example https://mytableauhost:8850. If no server is specified, https://<localhost | dnsname>:8850 is assumed.

--trust-admin-controller-cert

Optional.
Use this flag to trust the self-signed certificate on the TSM controller. For more information about certificate trust and CLI connections, see Connecting TSM clients.

-u, --username <user>

Required if no session is active, along with -p or --password.

Specify a user account. If you do not include this option, the command is run using credentials you signed in with.

tsm status

You can use the tsm status command to display the status of Tableau Server.

Synopsis

tsm status [global options]

Options

-v, --verbose

Optional.

Display status for every node in the Tableau Server cluster.

`tsm status` will return one of four potential statuses for a Tableau Server node:

- **RUNNING**: The node is running without error statuses for any service.
- **DEGRADED**: A primary service - such as the repository - is in an error state.
- **ERROR**: One or more services is in an error state.
- **STOPPED**: The node is stopped.
When running `tsm status` with the `--verbose` option, TSM will return a status for each individual service. There are four status messages:

- **is running:** The service is running.
- **status is unavailable:** The status is not returned - such as when processes are starting up.
- **is in an error state:** The process is running, but returning errors. This status indicates the process failed to install properly, or has not been configured.
- **is stopped:** The process is stopped.

Global options

- **-h, --help**
  
  Optional.

  Show the command help.

- **-p, --password <password>**

  Required, along with `-u` or `--username` if no session is active.

  Specify the password for the user specified in `-u` or `--username`.

  If the password includes spaces or special characters, enclose it in quotes:

  `--password 'my password'`

- **-s, --server <url_to_tsm>**

  Optional.

  Use the specified address for Tableau Services Manager. The URL must start with `https`, include port 8850, and use the server name not the IP address, for example

--trust-admin-controller-cert

Optional.

Use this flag to trust the self-signed certificate on the TSM controller. For more information about certificate trust and CLI connections, see Connecting TSM clients.

-u, --username <user>

Required if no session is active, along with -p or --password.

Specify a user account. If you do not include this option, the command is run using credentials you signed in with.

tsm stop

You can use the tsm stop command to stop Tableau Server.

Synopsis

tsm stop [option] [global options]

Option

--request-timeout <timeout in seconds>

Optional.

Wait the specified amount of time for the command to finish. Default value is 1800 (30 minutes).

Global options

-h, --help

Optional.
Show the command help.

-p, --password <password>

Required, along with -u or --username if no session is active.

Specify the password for the user specified in -u or --username.

If the password includes spaces or special characters, enclose it in quotes:

--password 'my password'

-s, --server <url_to_tsm>

Optional.

Use the specified address for Tableau Services Manager. The URL must start with https, include port 8850, and use the server name not the IP address, for example https://mytableauhost:8850. If no server is specified, https://<localhost | dnsname>:8850 is assumed.

--trust-admin-controller-cert

Optional.

Use this flag to trust the self-signed certificate on the TSM controller. For more information about certificate trust and CLI connections, see Connecting TSM clients.

-u, --username <user>

Required if no session is active, along with -p or --password.

Specify a user account. If you do not include this option, the command is run using credentials you signed in with.
You can use the `tsm topology` commands to prepare File Store nodes for safe removal or to put them back into read-write mode. You can also initiate a repository failover, get a list of nodes or ports, get the bootstrap configuration file required to add additional nodes to your cluster, remove nodes, and configure external repository.

- cleanup-coordination-service
- deploy-coordination-service
- external-services
  - list
  - repository
    - enable
    - disable -n nodeN
- failover-repository
- filestore
  - decommission
  - recommission
- list-nodes
- list-ports
- nodes
  - get-bootstrap-file
- remove-nodes
- set-node-role
- set-ports
- set-process
- toggle-coordination-service

**tsm topology cleanup-coordination-service**

Use the `tsm topology cleanup-coordination-service` command to remove the non-production Tableau Server Coordination Service ensemble after you deploy a new ensemble. This command removes the old Coordination Service instances on all nodes in the non-production Coordination Service ensemble and is required after you deploy a new
Coordination Service ensemble. To learn more about Coordination Service ensembles, see Deploy a Coordination Service Ensemble.

Synopsis

tsm topology cleanup-coordination-service [option] [global options]

Option

--request-timeout <timeout in seconds>

Optional.

Wait the specified amount of time for the command to finish. Default value is 2700 (45 minutes).

tsm topology deploy-coordination-service

You can use the tsm topology deploy-coordination-service command to deploy the Tableau Server Coordination Service. This command deploys a Coordination Service ensemble, which is a set of Coordination Service instances that run on specified nodes in your server cluster. To learn more about Coordination Service ensembles, including how many nodes in your cluster should have a Coordination Service instance, see Deploy a Coordination Service Ensemble.

Synopsis

tsm topology deploy-coordination-service --nodes <nodeID,nodeID,...> [option] [global-options]

Options

-n, --nodes <nodeID,nodeID,...>

Required.
Node IDs of nodes to include in the new Coordination Service ensemble, separated by commas. You can specify 1, 3, or 5 Coordination Service nodes, depending on the total number of nodes in your cluster. For more information, see The Coordination Service Quorum.

```
--request-timeout <timeout in seconds>
```

Optional.

Wait the specified amount of time for the command to finish. Default value is 2700 (45 minutes).

```
tsm topology external-services list
```

Use the `tsm topology external-service-list` command to get a the service that is used for Tableau Server External Repository. For example, if you have configured Tableau Server to use Amazon RDS, you will see the following message:

> These externally configured services are in use by Tableau Server:

```
- pgsql
```

**Synopsis**

```
tsm topology external-service list [global options]
```

**Option**

There are no options for this command.

```
tsm topology external-services repository enable
```

Use the `tsm topology external-services repository enable` command to configure Tableau Server to use an external repository. This command can be used during installation of a new Tableau Server to configure the external repository. If this command is run on an already existing and running Tableau Server, it will migrate the data from the local node to the external repository and configure Tableau Server to use the external repository after the migration is complete.
Synopsis

`tsm topology external-services repository enable -f <file-name>.json -c <ssl certificate file>.pem`

Options

- `-f <file name>`
  
  Required.
  
  Full path and file name where the configuration file is saved. For more information, see Re-Configure Tableau Server Repository.

- `-c <ssl certificate file>`
  
  Required.
  
  You must use SSL if you are using Amazon RDS for your external repository. Download the .pem file and specify the .pem file for use with this option. For more information on how to get the .pem file, see Using SSL to Encrypt a Connection to a DB Instance.

`tsm topology external-services repository disable -n nodeN`

Use the `tsm topology external-services repository disable` command to stop using the external repository and reconfigure the installation to use a local repository. This will migrate the data to a local repository and configure Tableau Server to use the local repository.

Synopsis

`tsm topology external-services repository disable -n nodeN`

Option

- `-n, --node-name <nodeID>`
  
  Required.
  
  Specifies the node ID of the node where the repository should be moved to.
This does not stop or delete the RDS instance. For more information on how to delete an RDS instance, see Deleting a DB Instance on AWS site.

Global options

-h, --help

Optional.

Show the command help.

-p, --password <password>

Required, along with -u or --username if no session is active.

Specify the password for the user specified in -u or --username.

If the password includes spaces or special characters, enclose it in quotes:

--password 'my password'

-s, --server <url_to_tsm>

Optional.

Use the specified address for Tableau Services Manager. The URL must start with https, include port 8850, and use the server name not the IP address, for example https://mytableauhost:8850. If no server is specified, https://<localhost | dnsname>:8850 is assumed.

--trust-admin-controller-cert

Optional.

Use this flag to trust the self-signed certificate on the TSM controller. For more information about certificate trust and CLI connections, see Connecting TSM clients.

-u, --username <user>
Required if no session is active, along with \(-p\) or \(--\text{password}\).

Specify a user account. If you do not include this option, the command is run using credentials you signed in with.

tsm topology failover-repository

You can use the \texttt{tsm topology failover-repository} to manually initiate a repository failover from the current active repository to the second, passive repository.

The \texttt{tsm topology failover-repository} command is persistent. The failover repository remains the active repository until you issue the command again, or, if Tableau Server is configured for it, until automatic failover occurs. If you have a preferred active repository configured, use the \texttt{--preferred} option to switch back to that repository. For more information about configuring a preferred active repository, see Tableau Server Repository. If Tableau Server is configured for high availability, failover of the repository is automatic when necessary. Use the failover-repository command to manually fail over the repository.

Synopsis

\texttt{tsm topology failover-repository --preferred | --target \<node_id\> [global options]}

Options

\texttt{-r, --preferred}

Required if -t or --target is not used.

Use the configured preferred node as the target for repository failover.

\texttt{--request-timeout <timeout in seconds>}

Optional.

Wait the specified amount of time for the command to finish. Default value is 1800 (30 minutes).
-t, --target <node_id>

   Required if -r or --preferred is not used.

   The node id of the target node onto which failover will occur. Find the node id by using the tsm topology list-nodes command.

**tsm topology filestore decommission**

You must use the tsm topology filestore decommission command to prepare a file store node or nodes for safe removal. This command puts the specified nodes into read-only mode and ensures there is no unique content on the specified nodes.

If decommissioning results in a single file store node, you must use the --override option or the decommission will fail.

**Synopsis**

```bash
tsm topology filestore decommission --nodes <nodeID,nodeID,...> [options] [global options]
```

**Options**

- -n, --nodes <nodeID,nodeID,...>

   Required.

   List of one or more nodes to decommission, specified by node ID and separated by commas.

- --delete-filestore

   Optional.

   Forces the removal of the file store, even if it has not been decommissioned. You should only use this option if the node the file store is on is in a error state and decommissioning cannot be done. Any unique files on the node will be permanently deleted.
-o, --override

Optional.

Overrides warnings or failures that would normally occur if removing the target File Store node would reduce the number of remaining file store nodes to one.

--request-timeout <timeout in seconds>

Optional.

Wait the specified amount of time for the command to finish. Default value is 1800 (30 minutes).

**tsm topology filestore recommission**

*Use the tsm topology filestore recommission command to revert any decommissioned nodes back to read-write mode.*

**Synopsis**

```
tsm topology filestore recommission --nodes <nodeID,nodeID,...> [global options]
```

**Options**

-n, --nodes <nodeID,nodeID,...>

Required.

List of one or more nodes to recommission, specified by node ID and separated by commas.

**tsm topology list-nodes**

Display the nodes in the cluster and (optionally) the services on each node.

**Synopsis**

```
tsm topology list-nodes [options] [global options]
```
Options

-\*\*-v, --verbose

Optional.

Shows each node ID, the node role (for more information, see set-node-role below), the node address, and the processes on each node.

tsm topology list-ports

Display the ports in the cluster.

Synopsis

tsm topology list-ports [options] [global options]

Options

--node-name <nodeID>

Optional.

Specify the node to list ports for.

--service-name

Optional.

Specify the service to list ports for.

tsm topology nodes get-bootstrap-file

You can use the tsm topology nodes get-bootstrap-file command to get the bootstrap file that is required to add a new node to the cluster.
**Important:** The bootstrap file contains a copy of the master keystore file used for encrypting the configuration secrets. The file can also embedded credentials which are valid for a predetermined amount of time (see `tabadmincontroller.auth.expiration.minutes`) and serve as a session cookie. We strongly recommend that you take additional measures to secure the bootstrap file using mechanisms as described in Securing secrets for import and export operations.

### Synopsis

```
tsm topology nodes get-bootstrap-file --file <path\file>.json
```

### Options

- `-f, --file <file>`
  - Required.
  - Full path and file name where the configuration file will be saved. If a duplicate file exists it will be overwritten.

- `-ne, --no-embedded-credential`
  - Optional.
  - Added in version 2019.3.

  By default embedded credentials are included in the bootstrap file. Use this option if credentials should not be included in the bootstrap file. Embedded credentials are temporary, and expire based on the value of the `tabadmincontroller.auth.expiration.minutes` configuration key, by default 120 minutes.
**Note:** You can disable the ability to include embedded credentials at the server level, using a configuration option. For more information, see features.PasswordlessBootstrapInit.

```
tsm topology remove-nodes
```

Remove nodes from the cluster.

To complete removal of a node, you also must run the `tsm pending-changes apply` command. Some scenarios require that you move or redeploy processes before removing nodes. See Remove a Node.

If you remove a node and want to re-add it to the cluster, you need to first run the `obliterate` script to clean Tableau off it, then reinstall the node using the normal process for adding a new node. For more information, see Remove Tableau Server from Your Computer and Install and Configure Additional Nodes.

**Note:** To remove a node from a cluster it must have been configured with a process at some point in the past. If you are removing a node on which you've not configured any processes, then you must add a process on it, run `tsm pending-changes apply`, and then remove the node.

**Synopsis**

```
tsm topology remove-nodes --nodes <nodeID,nodeID,...>
[global options]
```

**Options**

```
-n, --nodes <nodeID,nodeID,...>
```

Required.
Specify the node or nodes to remove. If specifying multiple nodes, separate node IDs with a comma.

**tsm topology set-node-role**

Set the Backgrounder and Extract Queries node roles. This determines the type of tasks that will be performed on the nodes. The following node roles are useful if you have a multi-node cluster and requires Add-on licenses. For more information, see Workload Management through Node Roles.

**Note:** Making configurations to node roles require a restart of the server and will require some downtime. For more information, see tsm pending-changes.

**Synopsis**

```
 tsm topology set-node-role [options] [global options]
```

**Options**

```
-n, --nodes <nodeID,nodeID,...>
```

Required.

List of one or more nodes to set node roles for, specified by node ID and separated by commas and without spaces between nodes.

```
-r --role <all-jobs,flows,no-flows,extract-refreshes,subscriptions,extract-refreshes-and-subscriptions,no-extract-refreshes,no-subscriptions,no-extract-refreshes-and-subscriptions,extract-queries>
```

Required

Sets the role for the nodes specified. The valid values for this option are:
- all-jobs: Backgrounder will run all types of jobs.
- flows: Backgrounder will run only flow run jobs.
- no-flows: Backgrounder will not run flow run jobs.
- extract-refreshes: Backgrounder will run only extract refresh jobs. This includes, incremental refreshes, full refreshes, encryption and decryption of all extracts including extracts that flow outputs generate.
- subscriptions: Backgrounder will run only subscription jobs.
- extract-refreshes-and-subscriptions: Backgrounder will run extract-refreshes, encryption and decryption of all extracts including extracts that flow outputs create, and subscription jobs.
- no-extract-refreshes: Backgrounder will run all jobs except extract-refreshes, extract encryption and decryption including extracts created from flow outputs.
- no-subscriptions: Backgrounder will run all jobs except subscriptions.
- no-extract-refreshes-and-subscriptions: Backgrounder will run all jobs except extract-refreshes, encryption and decryption of all extracts including extracts created from flow outputs, and subscriptions.
- extract-queries: The nodes selected will run as all-jobs and will prioritize the processing of extract queries.

tsm topology set-ports

Set the ports for a service instance.

Synopsis

tsm topology set-ports --node-name <nodeID> --port-name <port_name> --port-value <port_value> [options] [global options]

Options

-i, --instance <instance_id>
Optional.

Specifies the instance id of the service. Defaults to 0 (zero) if not specified.

-n, --node-name <nodeID>

Required.

Specifies the node ID of the node.

-pn, --port-name <port_name>

Required.

The name of the port to be set, in this format: service_name:port_type. If no port type is specified, the primary port is assumed. For port name syntax, see Dynamically mapped ports.

-pv, --port-value <port_value>

Required.

The port to set.

-r, --restart

Optional.

Suppress the prompt for restart and restart Tableau Server when necessary.

tsm topology set-process

Set the number of instances of a process on a node. If a node already has the specified process, the number is updated to match the specified count.

You can only set one process at a time. If you specify more than one process, any process after the first one will be silently ignored.
Note: For a complete list of process names, see Tableau Server Processes.

Synopsis

```
tsm topology set-process --count <process_count> --node <nodeID> --process <process_name> [global options]
```

Options

- `-c, --count <process_count>`
  
  Required.
  
  The process count (number of instances) to set.

- `-n, --node <nodeID>`
  
  Required.
  
  Specifies the node ID of the node on which to set the process.

- `-pr, --process <process_name>`
  
  Required.
  
  The name of the process to be set.

```tsm topology toggle-coordination-service```

You can use the `tsm topology toggle-coordination-service` command to switch between coordination service ensembles. To learn more about Coordination Service ensembles, see Deploy a Coordination Service Ensemble.
Tableau Server on Linux Administrator Guide

Synopsis

tsm topology toggle-coordination-service [option]
[global options]

Option

--request-timeout <timeout in seconds>

Optional.

Wait the specified amount of time for the command to finish. Default value is 1800 (30 minutes).

tsm user-identity-store

You can use the tsm user-identity-store commands to modify the settings of the identity store for Tableau Server after the initial configuration.

The initial configuration of the identity store is part of the installation process. See Configure Initial Node Settings.

For introduction to identity store concepts, see Identity Store.

For LDAP/Active Directory configuration reference table, see LDAP Configuration Reference.

- get-group-mappings
- get-user-mappings
- list
- set-connection
- set-group-mappings
- set-user-mappings
- verify-group-mappings
- verify-user-mappings
tsm user-identity-store get-group-mappings [options]
Displays identity store group mappings.

Synopsis

```
 tsm user-identity-store get-group-mappings [global options]
```

tsm user-identity-store get-user-mappings [options]
Displays identity store user mappings.

Synopsis

```
 tsm user-identity-store get-user-mappings [global options]
```

tsuser-identity-store list [options]
Lists user-identity configuration.

Synopsis

```
 tsm user-identity-store list [options] [global options]
```

Options

- `v, --verbose`
  Optional.
  Lists all configuration parameters.

tsuser-identity-store set-connection [options]
Sets identity store connection parameters.

Synopsis

```
 tsm user-identity-store set-connection --kerbkeytab <kerb-keytab> [options] [global options]
```

Options

-\(b\), --bind <username and password | Kerberos>
  Optional.
  Set LDAP bind type.

-\(d\), --domain <domain>
  Optional.
  Domain name.

-\(hn\), --hostname <hostname>
  Optional.
  Hostname value.

-\(kc\), --kerbconfig <kerbconfig>
  Optional.
  Kerberos configuration file path.

-\(kp\), --kerbprincipal <kerbprincipal>
  Optional.
  Kerberos Principal.

-\(kt\), --kerbkeytab <kerbkeytab>
  Required.
  Kerberos keytab file path.
-l,--port <port>
  Optional.
  Set LDAP Port value.

-lp,--ldappassword <ldappassword>
  Optional.
  LDAP Password.

-lu,--ldapusername <ldapusername>
  Optional.
  Set LDAP Username value.

-n,--nickname <nickname>
  Optional.
  NetBIOS name (nickname).

**tsm user-identity-store set-group-mappings [options]**

Sets identity store group mappings and configures LDAP directories that implement an arbitrary or custom schema.

**Synopsis**

```
 tsm user-identity-store set-group-mappings [options]
[global options]
```

**Options**

-b,--basefilter <groupbasefilter>
  Optional.
Set group BaseFilter value.

-\( \text{cn,--classnames <group_classnames>} \)

Optional.

Override default user classname values (contains "group" string) with the values you set here. You can provide multiple classnames separated by commas.

-\( \text{d,--description <description>} \)

Optional.

Group description.

-\( \text{e,--groupemail <groupemail>} \)

Optional.

Group email value.

-\( \text{m,--member <member>} \)

Optional.

Set the group members.

-\( \text{n,--groupname <groupname>} \)

Optional.

Name of the group.

tsm user-identity-store set-user-mappings [options]

Sets identity store user mappings and configures LDAP directories that implement an arbitrary or custom schema.
Synopsis

tsm user-identity-store set-user-mappings --certificate <certificate> [options] [global options]

Options

-c, --certificate <certificate>

Optional.

Users' certificate file location.

-cn, --classnames <user_classnames>

Optional.

Override default user classname values ("user" and "inetOrgPerson") with the values you set here. You can provide multiple classnames separated by commas.

-dn, --displayname <displayname>

Optional.

Display name of the user.

-e, --email <email>

Optional.

Users' email address.

-jp, --jpegphoto <jpegfile>

Optional.

Users' jpeg image location.

-m, --memberof <groupname>
Tableau Server on Linux Administrator Guide

Optional.

Group that the user is a member of.

-t, --thumbnail <thumbnail>

Optional.

Users' thumbnail location.

-ub, --basefilter <userbasefilter>

Optional.

Users' BaseFilter.

-uu, --ldapusername <ldapusername>

Optional.

User name.

tsm user-identity-store verify-group-mappings [options]

Validates configuration for LDAP group mapping.

Synopsis

tsm user-identity-store verify-group-mappings --verify <group_name> [global options]

Options

-v, --verify <group_name>

Optional.

Name of group to search for.
tsm user-identity-store verify-user-mappings [options]

Validates configuration for LDAP user mapping.

Synopsis

`tsm user-identity-store verify-user-mappings --verify <user_name> [global options]`

Options

- **-v, --verify <user_name>**
  
  Optional.

  Name of user to search for.

Global options

- **-h, --help**
  
  Optional.

  Show the command help.

- **-p, --password <password>**
  
  Required, along with -u or --username if no session is active.

  Specify the password for the user specified in -u or --username.

  If the password includes spaces or special characters, enclose it in quotes:

  --password 'my password'

- **-s, --server <url_to_tsm>**
  
  Optional.
Use the specified address for Tableau Services Manager. The URL must start with https, include port 8850, and use the server name not the IP address, for example https://mytableauhost:8850. If no server is specified, https://<localhost | dnsname>:8850 is assumed.

--trust-admin-controller-cert

Optional.

Use this flag to trust the self-signed certificate on the TSM controller. For more information about certificate trust and CLI connections, see Connecting TSM clients.

-u, --username <user>

Required if no session is active, along with -p or --password.

Specify a user account. If you do not include this option, the command is run using credentials you signed in with.

tsm version

You can use the tsm version command to get the versions of TSM and Tableau Server.

Synopsis

tsm version [global options]

Global options

-h, --help

Optional.

Show the command help.

-p, --password <password>

Required, along with -u or --username if no session is active.
Specify the password for the user specified in -u or --username.

If the password includes spaces or special characters, enclose it in quotes:

--password 'my password'

-s, --server <url_to_tsm>

Optional.

Use the specified address for Tableau Services Manager. The URL must start with https, include port 8850, and use the server name not the IP address, for example https://mytableauhost:8850. If no server is specified, https://<localhost | dnsname>:8850 is assumed.

--trust-admin-controller-cert

Optional.

Use this flag to trust the self-signed certificate on the TSM controller. For more information about certificate trust and CLI connections, see Connecting TSM clients.

-u, --username <user>

Required if no session is active, along with -p or --password.

Specify a user account. If you do not include this option, the command is run using credentials you signed in with.

tsm File Paths

Certain tsm commands read files from or write files to default locations. These default locations are determined by basefilepath variables defined for each command. You can use tsm to view the current value of the variables, and to change the locations.
Default locations for files

During the `tsm` maintenance backup, restore, send-logs, and `ziplogs` processes, and the `tsm` sites export and sites import processes, Tableau Server uses default locations for the files created or used by these commands.

By default:

- **tsm maintenance commands:**
  - **backup**—The backup `.tsbak` file is generated in:
    
    ```bash
    /var/opt/tableau/tableau_server-
data/tabsvc/files/backups
    ```
  - **restore**—The restore process restores a backup file from:
    
    ```bash
    /var/opt/tableau/tableau_server-
data/tabsvc/files/backups
    ```
  - **send-logs**—The send-logs sends the logs file from:
    
    ```bash
    /var/opt/tableau/tableau_server-
data/tabsvc/files/backups
    ```
  - **ziplogs**—The ziplogs file is generated in:
    
    ```bash
    /var/opt/tableau/tableau_server/data/tabsvc/files/log-archives
    ```

- **tsm sites**
  - **export**—The export `.zip` file is generated to the following directory:
    
    ```bash
    /var/opt/tableau/tableau_server-
data/tabsvc/files/siteexports
    ```
• import—During the import process, Tableau Server looks for files in:

/var/opt/tableau/tableau_server-
/data/tabsvc/files/siteimports

Get the current file location

You can see the current file location for a specific command using `tsm configuration get`:

• For tsm maintenance commands:
  • backup, restore, and send-logs:
    `tsm configuration get -k basefilepath.backuprestore`
  • ziplogs:
    `tsm configuration get -k basefilepath.log_archive`

• For tsm sites commands:
  • export
    `tsm configuration get -k basefilepath.site_export.-exports`
  • import
    `tsm configuration get -k basefilepath.site_import.-exports`

Change the current file location

You can change the expected file locations using the `tsm configuration set` command to update the `basefilepath` variables.
Changing a `basefilepath` variable does not move existing files from the original directory to the new directory. If you want existing backup, restore, log files, or site export or import files to reside in the new directory you specify, you must move them manually. You are responsible for creating the new location and for setting the correct permissions to allow tsm access to any files that will be placed there, and to the directory structure containing those files. For more information about permissions and tsm, see Files and Permissions in TSM.

- **For tsm maintenance commands:**
  - To change the backup, restore, or send-logs directory, run the following command:
    
    ```
    tsm configuration set -k basefilepath.backuprestore -v 
    "/new/directory/path"
    ```

  - To change the ziplogs directory:
    
    ```
    tsm configuration set -k basefilepath.log_archive -v 
    "/new/directory/path"
    ```

- **For tsm sites commands:**
  - To change the sites export directory:
    
    ```
    tsm configuration set -k basefilepath.site_export.-exports -v 
    "/new/directory/path"
    ```

  - To change the sites import directory:
    
    ```
    tsm configuration set -k basefilepath.site_import.-exports -v 
    "/new/directory/path"
    ```

After you change a default file location you need to do the following:
1. Apply pending changes:

   tsm pending-changes apply

2. Stop Tableau Server:

   tsm stop

3. Restart the TSM Controller (as tableau system account):

   sudo su -l tableau -c "systemctl --user restart tabadmincontroller_0.service"

4. Wait several minutes for the controller to restart. You can confirm the controller has restarted with this command:

   tsm status -v

   When you can run that command and the Tableau Server Administration Controller is listed as 'running' the controller has restarted.

5. Start Tableau Server:

   tsm start

Migrate from Tabadmin to the TSM CLI

The Tableau Services Manager (TSM) command-line interface (CLI) replaces the tabadmin CLI in Tableau Server on Linux, and in Tableau Server on Windows version 2018.2. This page maps tabadmin commands to TSM commands to help you to migrate to the TSM CLI.

To learn more about the TSM CLI, see tsm Command Line Reference.

Tabadmin commands with a corresponding TSM CLI command

The following table shows which tabadmin commands correspond to commands available in the TSM CLI.

<table>
<thead>
<tr>
<th>Command Description</th>
<th>Tabadmin Command(s)</th>
<th>Comparable TSM CLI Command</th>
</tr>
</thead>
<tbody>
<tr>
<td>Activate a license</td>
<td>tabadmin activate --activate</td>
<td>tsm licenses activate</td>
</tr>
<tr>
<td>Deactivate licenses</td>
<td>tabadmin activate --return</td>
<td>tsm licenses deactivate</td>
</tr>
<tr>
<td>Activate a trial license</td>
<td>tabadmin activate --trial</td>
<td>tsm licenses activate --trial</td>
</tr>
<tr>
<td>Create a backup of the data managed by Tableau Server</td>
<td>tabadmin backup</td>
<td>tsm maintenance backup</td>
</tr>
<tr>
<td>Clear the server cache</td>
<td>tabadmin clearcache</td>
<td>tsm maintenance cleanup -r</td>
</tr>
<tr>
<td>Clean up temporary files and old log files</td>
<td>tabadmin cleanup</td>
<td>tsm maintenance cleanup</td>
</tr>
<tr>
<td>Update the</td>
<td>tabadmin configure</td>
<td>tsm pending-changes</td>
</tr>
</tbody>
</table>

A backup created using TSM does not include any server configuration data. There is no option to include server configuration data.

Note: This command was added in version 10.5.1
<table>
<thead>
<tr>
<th>Server Configuration with any changes you've made</th>
<th>Apply</th>
</tr>
</thead>
<tbody>
<tr>
<td>Customize the server name and logos</td>
<td>tabadmin customize</td>
</tr>
<tr>
<td>Enable access to the repository</td>
<td>tabadmin dbpass</td>
</tr>
<tr>
<td>Disable access to the repository</td>
<td>tabadmin dbpass --disable</td>
</tr>
<tr>
<td>Set a file store instance to read-only mode</td>
<td>tabadmin decommission</td>
</tr>
<tr>
<td>Delete one or more Web Data Connectors (WDCs) from Tableau Server</td>
<td>tabadmin delete_web-dataconnector</td>
</tr>
<tr>
<td>Add a Web Data Connector (WDC) to Tableau Server</td>
<td>tabadmin import_web-dataconnector and</td>
</tr>
<tr>
<td>Note: TSM does not support importing WDCs, instead it lets you add WDCs to a safe list (or &quot;whitelist&quot;). To learn more, see Web Data Connectors in Tableau Server.</td>
<td>tsm data-access web-data-connectors add</td>
</tr>
<tr>
<td>Task</td>
<td>Command</td>
</tr>
<tr>
<td>-----------------------------------------------------------</td>
<td>--------------------</td>
</tr>
<tr>
<td>List Web Data Connectors (WDCs) used by Tableau Server</td>
<td>tabadmin list_web-dataconnectors</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>Export a site from Tableau Server</td>
<td>tabadmin exportsite</td>
</tr>
<tr>
<td>Initiate a repository failover</td>
<td>tabadmin fail-overrepository</td>
</tr>
<tr>
<td>Get a configuration option</td>
<td>tabadmin get</td>
</tr>
<tr>
<td>Get the OpenID redirect URL</td>
<td>tabadmin get_openid_redirect_url</td>
</tr>
<tr>
<td>Import site .csv files into Tableau Server</td>
<td>tabadmin importsite</td>
</tr>
<tr>
<td>Import a site into Tableau Server using .csv files</td>
<td>tabadmin importsite_verified</td>
</tr>
<tr>
<td>Display license information for Tableau Server</td>
<td>tabadmin licenses</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>Move a file store from</td>
<td>tabadmin recommission</td>
</tr>
<tr>
<td>Task</td>
<td>Tabadmin Command</td>
</tr>
<tr>
<td>----------------------------------------------------------------------</td>
<td>-----------------------------------------</td>
</tr>
<tr>
<td>Regenerate internal security tokens</td>
<td>tabadmin regenerate_internal_tokens</td>
</tr>
<tr>
<td>Register Tableau Server</td>
<td>tabadmin register</td>
</tr>
<tr>
<td>Rebuild the search index for Tableau Server</td>
<td>tabadmin reindex</td>
</tr>
<tr>
<td>Reset the Tableau Server administrator account</td>
<td>tabadmin reset</td>
</tr>
<tr>
<td>Stop and restart all Tableau Server processes</td>
<td>tabadmin restart</td>
</tr>
<tr>
<td>Restore from a Tableau Server backup file</td>
<td>tabadmin restore</td>
</tr>
<tr>
<td>- The restore command does not restore any server configuration data. This is true whether you are using a backup created with TSM or a backup created with tabadmin.</td>
<td></td>
</tr>
<tr>
<td>Set a configuration option</td>
<td>tabadmin set</td>
</tr>
</tbody>
</table>

**Note:** Added in version 2018.1.
Activate or suspend a site  
**Command**  
**Tabadmin Command**  
**Notes**  
Start all Tableau Server processes  
**Command**  
**Tabadmin Command**  
**Notes**  
Get the status of Tableau Server and server processes  
**Command**  
**Tabadmin Command**  
**Notes**  
Stop all Tableau Server processes  
**Command**  
**Tabadmin Command**  
**Notes**  
Create an archive (.zip) file with Tableau Server log files  
**Command**  
**Tabadmin Command**  
**Notes**

**Tabadmin commands with no corresponding TSM CLI command**

The following table lists the tabadmin commands for which a comparable TSM CLI command is not available.

<table>
<thead>
<tr>
<th>Command Description</th>
<th>Tabadmin Command</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add or remove a user from the system administrator group</td>
<td>tabadmin administrator</td>
<td>You can use the Tableau Server REST API Add User to Group and Remove User from Group methods to add or remove a user from the system administrator group.</td>
</tr>
<tr>
<td>Create a new key to encrypt sensitive information stored in the repository</td>
<td><code>tabadmin assetkeys</code></td>
<td>Use the <code>tsm security regenerate-internal-tokens</code> command to create or regenerate secrets and master keys.</td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>Specify whether Tableau Server starts at system start-up time</td>
<td><code>tabadmin autostart</code></td>
<td>Tableau Server returns to the state it was in prior to a system restart. If it was running, it will restart. If it was stopped it will be stopped after the system starts.</td>
</tr>
<tr>
<td>Identify a second server node for backup</td>
<td><code>tabadmin fail-overprimary</code></td>
<td>TSM does not have primary nodes, so a TSM equivalent to this command is not needed.</td>
</tr>
<tr>
<td>Manage credentials for delegated data access on Tableau Server</td>
<td><code>tabadmin manage_global_credentials</code></td>
<td>We recommend that you use Kerberos delegation to Apache Impala for global credential management. To learn more, see Kerberos and <a href="https://community.tableau.com/s/kerberos-and-enable-kerberos-delegation">Enable Kerberos Delegation for Hive/Impala</a> in the Tableau Community.</td>
</tr>
<tr>
<td>Reset the password for a Tableau Server account</td>
<td><code>tabadmin passwd</code></td>
<td>If your server uses local authentication, you can use the Tableau Server REST API <code>Update User</code> method to reset the password for a user account.</td>
</tr>
<tr>
<td>Reset binding between Tableau Server user ID and Open ID Connect identity provider</td>
<td><code>tabadmin reset_openid_sub</code></td>
<td>—</td>
</tr>
</tbody>
</table>
Determine whether your environment meets the minimum requirements to run Tableau Server:

```
tabadmin validate
```

Verify that a backup of the Tableau Server repository will restore successfully:

```
tabadmin verify_database
```

The `tsm maintenance backup` command automatically verifies that a backup will restore correctly unless you use the `--skip-verification` parameter.

Prepare VizQL processes for fast load times after a Tableau Server restart:

```
tabadmin warmup
```

The `tabadmin warmup` command is no longer necessary, as Tableau Server is now optimized to automatically provide fast load times after a server restart.

---

**Entity Definitions and Templates**

You can use the entity definitions and templates with the `tsm settings import` command to configure various Tableau Server settings.
Configuration File Example

This article provides an example of a complete JSON configuration file, with gatewaySettings and identityStore entities specified. In addition, a configuration key sets the gateway timeout to 900 seconds.

Your configuration file will look different depending on the options you need to set.

You might set multiple .json configuration files during installation. To set the values for each file in Tableau Server, you run the following command, once for each configuration file:

```
tsm settings import -f path-to-file.json
```

After you set the configuration files, run tsm pending-changes apply to apply the changes from all of the .json files you’ve set.

```
{
  "configEntities": {
    "gatewaySettings": {
      "_type": "gatewaySettingsType",
      "port": 80,
      "firewallOpeningEnabled": true,
      "sslRedirectEnabled": true,
      "publicHost": "localhost",
      "publicPort": 80
    },
    "identityStore": {
      "_type": "identityStoreType",
      "type": "local",
      "domain": "example.lan",
      "nickname": "EXAMPLE"
    }
  },
  "configKeys": {
    "gateway.timeout": "900"
  }
}
```
Entities vs keys

As shown in the example above, there are two classes of configuration parameters: configEntities and configKeys.

configEntities

Certain types of configuration are done through entity sets that map to specific scenarios, such as the identity store and gateway configurations. When you pass a set of configEntities with the `tsm settings import -f path-to-file.json` command, TSM validates the configuration. If values passed are invalid, TSM will provide an error. This enables you to make changes during the configuration process, rather than experience a configuration failure at initialization or run time.

Entities can be set only by including a configEntities block in a .json file.

**Important:** All files that are referenced in configEntities must be located on the local computer. Do not specify UNC paths.

configKeys

Entities cover only a small portion of the configuration values that can be set. Hundreds of keys correspond to parameters stored in .yml files. Tableau Server uses these parameters to store all of the configuration information for all services.

You can set individual keys with the `tsm configuration` command. But during deployment, setting them along with other configuration scenarios in JSON files, as shown above, is more convenient.

Unlike configEntities, configKeys are not validated.
Note: We do not recommend setting parameters that are not documented in tsm configuration set Options.

gatewaySettings Entity

You must configure the gateway settings for the Tableau Server computer.

Use the configuration file template below to create a json file. After you have filled in the options with the appropriate values, pass the json file and apply settings with the following commands:

```
tsm settings import -f /path/to/file.json
tsm pending-changes apply
```

The `pending-changes apply` command displays a prompt to let you know this will restart Tableau Server if the server is running. The prompt displays even if the server is stopped, but in that case there is no restart. You can suppress the prompt using the `--ignore-prompt` option, but this does not change the restart behavior. For more information, see `tsm pending-changes apply`.

Gateway settings

The gateway settings in the template below specify the HTTP settings for Tableau Server. We recommend using SSL/TLS. Tableau Server is hard-coded to use port 443 for SSL/TLS. Therefore, if you enable SSL, you do not need to update the gatewaySettings entity.

Configuration template

Use this template to configure the gateway settings.

All entity options are case sensitive.
For more explanation about configuration files, entities, and keys see Configuration File Example.

```json
{
    "configEntities": {
        "gatewaySettings": {
            
            "_type": "gatewaySettingsType",
            "port": 80,
            "sslRedirectEnabled": true,
            "publicHost": "localhost"
        }
    }
}
```

Configuration file reference

This table includes all of the options that can be included with the "gatewaySettings" entity set.

_type

Required.

Value: "gatewaySettingsType"

Do not change.

port

Specifies HTTP port. Default is port 80.

sslRedirectEnabled

Options: true or false.

publicHost

Specifies host name for http/s service.
trustedIPs

Specifies trusted IP addresses that communicate with Tableau Server. Trusted IP addresses include upstream proxy servers and servers that are used for trusted authentication with Tableau Server. See Configuring Proxies for Tableau Server and Add Trusted IP Addresses or Host Names to Tableau Server.

If you are running Tableau Server in a cluster then all other nodes of the cluster will automatically be included in the corresponding configuration file that this entity updates. Therefore, if you specify a new value for trustedIPs, then you must include the IP addresses for the other nodes in the value.

This option takes a list of strings, which requires passing each IP or host in quotes, separated by a comma (no space) and within brackets. For example:

"["192.168.1.101","192.168.1.102","192.168.1.103"] or ["web-serv1","webserv2","webserv3"].

trustedHosts

Specifies trusted IP addresses that communicate with Tableau Server. Typically, this value contains a list of upstream proxy servers. The values in trustedHosts are used to determine client request targets.

If you are running Tableau Server in a cluster then all other nodes of the cluster will automatically be included in the corresponding configuration file that this entity updates. Therefore, if you specify a new value for trustedIPs, then you must include the IP addresses for the other nodes in the value.

This option takes a list of strings, which requires passing each IP or host in quotes, separated by a comma (no space) and within brackets. For example:

"["192.168.1.101","192.168.1.102","192.168.1.103"] or ["web-serv1","webserv2","webserv3"].

identityStore Entity

Tableau Server requires an identity store to store user and group information. Review Authentication and Identity Store topics before configuring the identity store for the first
After you have installed the identity store on Tableau Server, you cannot change it without reinstalling the server.

All entity options are case sensitive.

Before you begin

Review the following information:

- If you plan to enable OpenID Connect then you must configure the local identity store.

- If you will not be using the local identity store, then you will be using some version of LDAP. In this case, work with your directory/LDAP administrator to configure Tableau Server for your LDAP schema and bind requirements.

- Tableau Server configuration is optimized for Active Directory. If you are installing into Active Directory, we recommend configuring the identity store with Configure Initial Node Settings. Alternatively, if you are configuring with TSM CLI, use the LDAP - Active Directory template in this topic to configure identity store. For a full accounting of LDAP configuration options, see LDAP Configuration Reference.

- LDAP bind is independent of user authentication. For example, you can configure Tableau Server to use simple bind to authenticate to the LDAP directory and then configure Tableau Server to authenticate users with Kerberos after installation.

- Do not connect to LDAP with simple bind over a unsecured connection. We recommend LDAPS for simple bind. See LDAP over SSL.

- To use Kerberos authentication for the Tableau Server service, then you'll need a keytab file for GSSAPI bind, as described in the sections below. See also, Understanding Keytab Requirements. In the context of Kerberos, GSSAPI bind is all you need during the base installation of Tableau Server. After you install the server, you can then configure Kerberos for user authentication and Kerberos delegation to data sources.

- In this topic, we make the distinction between LDAP (the protocol for connecting to
directory services) and an **LDAP server** (an implementation of a directory service). For example, **slapd** is an LDAP server that is part of the OpenLDAP project.

- We recommend validating the LDAP configuration before initializing the server, see Configure Initial Node Settings.

Use one of the configuration file templates below to create a json file. After you have filled in the options with the appropriate values, pass the json file and apply settings with the following commands:

```
 tsm settings import -f path-to-file.json
 tsm pending-changes apply
```

The `pending-changes apply` command displays a prompt to let you know this will restart Tableau Server if the server is running. The prompt displays even if the server is stopped, but in that case there is no restart. You can suppress the prompt using the `--ignore-prompt` option, but this does not change the restart behavior. For more information, see `tsm pending-changes apply`.

**Configuration templates**

Select an identity store configuration template to edit:

- Local
- LDAP - Active Directory
- OpenLDAP - GSSAPI Bind
- OpenLDAP - Simple Bind

For more explanation about configuration files, entities, and keys see Configuration File Example.

Consider using the **Tableau Identity Store Configuration Tool** to generate your LDAP json configuration file. The tool itself is not supported by Tableau. However, using a JSON file created by the tool instead of creating a file manually does not change the supported status of your server.
Local

Configure local as the identity store type if your organization does not already have an Active Directory or LDAP server for user authentication. When you select local as the identity store type, you use Tableau Server to create and manage users.

```json
{
   "configEntities": {
      "identityStore": {
         "_type": "identityStoreType",
         "type": "local"
      }
   }
}
```

**Important**

The LDAP configuration templates below are examples. The templates, as presented, will not configure LDAP connectivity in your organization. You must work with your directory administrator to edit the LDAP template values for a successful deployment.

Additionally, all files that are referenced in configEntities must be located on the local computer. Do not specify UNC paths.

LDAP - Active Directory

Configure this option to connect to Active Directory with GSSAPI (Kerberos) bind. Tableau Server uses the LDAP protocol to connect to Active Directory. GSSAPI (Kerberos) bind is used to authenticate Tableau Server service to Active Directory. Tableau Server includes support for Active Directory schema. Therefore, if you set the "directoryServiceType" option to "activedirectory" then you do not need to provide schema info in the "identityStoreSchemaType" option.
If you are installing Tableau Server for Linux into Active Directory, and the computer where you are installing Tableau Server is already joined to the domain, then the computer will already have a Kerberos configuration file and a keytab file. Strictly speaking, you can use these files for GSSAPI bind, but we don't recommend using them. Instead, contact your Active Directory administrator and request a keytab specifically for the Tableau Server service.

```
{
    "configEntities": {
        "identityStore": {
            "_type": "identityStoreType",
            "type": "activedirectory",
            "domain": "your-domain.lan",
            "nickname": "YOUR-DOMAIN-NICKNAME",
            "directoryServiceType": "activedirectory",
            "bind": "gssapi",
            "kerberosKeytab": "<path to local key tab file>",
            "kerberosConfig": "/etc/krb5.conf",
            "kerberosPrincipal": "your-principal@YOUR.DOMAIN"
        }
    }
}
```

We recommend binding to Active Directory with GSSAPI. However, you can connect with simple bind and LDAPS. To connect with simple bind, change bind to simple, remove the three Kerberos entities, and add the port/sslPort, username, and password options. The following example shows Active Directory with simple bind json.

```
{
    "configEntities": {
        "identityStore": {
            "_type": "identityStoreType",
            "type": "activedirectory",
            "domain": "your-domain.lan",
            "nickname": "YOUR-DOMAIN-NICKNAME",
            "directoryServiceType": "activedirectory",
            "bind": "simple",
            "username": "your-username",
            "password": "your-password"
        }
    }
}
```
OpenLDAP - GSSAPI bind

Use the template below to configure OpenLDAP with GSSAPI bind. Do not use this template if your organization is running Active Directory. If you are installing into Active Directory, use the template above, LDAP - Active Directory.

In most cases, organizations that use OpenLDAP with GSSAPI (Kerberos) will use a keytab file to store credentials. In the following example, a keytab file is used for authentication credentials.

However, you can provide credentials through the `username` and `password` entities.

You can also specify both a keytab and a username and password pair. In this case, Tableau Server will attempt to use the keytab, but if authentication fails for any reason it will fallback and use the username and password credentials.

```json
{
  "configEntities": {
    "identityStore": {
      
      
      "_type": "identityStoreType",
      "type": "activedirectory",
      "domain": "your-domain.lan",
      "nickname": "YOUR-DOMAIN-NICKNAME",
      "directoryServiceType": "openldap",
      "bind": "gssapi",
      "kerberosKeytab": "<path to local key tab file>"
    }
  }
}
```
Tableau Server on Linux Administrator Guide

"kerberosConfig": "/etc/krb5.conf",
"kerberosPrincipal": "your-principal@YOUR.DOMAIN",
"identityStoreSchemaType": {
    "userBaseFilter": "(objectClass=inetOrgPerson)",
    "userName": "user",
    "userDisplayName": "displayname",
    "userEmail": "email",
    "userCertificate": "certificate",
    "userThumbnail": "thumbnail",
    "userJpegPhoto": "photo",
    "groupBaseFilter": "(objectClass=groupofNames)",
    "groupName": "groupname",
    "groupEmail": "groupemail",
    "groupDescription": "groupdescription",
    "member": "member",
    "distinguishedNameAttribute": "",
    "serverSideSorting": "",
    "rangeRetrieval": "",
    "userClassNames": ["inetOrgPerson","someClass2"],
    "groupClassNames": ["groupOfUniqueNames1","groupOfUniqueNames2"]
}
Tableau Server on Linux Administrator Guide

"nickname": "",
"hostname": "optional-ldap-server",
"port": "389",
"directoryServiceType": "openldap",
"bind": "simple",
"username": "cn=username,dc=your,dc=domain",
"password": "password",
"identityStoreSchemaType": {
    "userBaseFilter": "(objectClass=inetOrgPerson)",
    "userUsername": "user",
    "userDisplayName": "displayname",
    "userEmail": "email",
    "userCertificate": "certificate",
    "userThumbnail": "thumbnail",
    "userJpegPhoto": "photo",
    "groupBaseFilter": "(objectClass=groupofNames)",
    "groupName": "groupname",
    "groupEmail": "groupemail",
    "groupDescription": "groupdescription",
    "member": "member",
    "distinguishedNameAttribute": "",
    "serverSideSorting": "",
    "rangeRetrieval": "",
    "userClassNames": ["inetOrgPerson","someClass2"],
    "groupClassNames": ["groupOfUniqueNames1","groupOfUniqueNames2"]
}
}
Configuration template reference

Shared identity store options

type
Where you want to store user identity information. Either local or activedirectory. (If you want to connect to any LDAP server, select activedirectory.)

domain
The domain of the computer where you installed Tableau Server.

nickname
The nickname of the domain. This is also referred to as the NetBIOS name in Windows environments.
The nickname option is required for all LDAP entities. If your organization does not require a nickname/NetBIOS, then pass a blank key, for example: "nickname": "".

LDAP GSSAPI bind options

directoryservicetype
The type of directory service that you want to connect to. Either activedirectory or openldap.

kerberosConfig
The path to the Kerberos configuration file on the local computer. If you are installing into Active Directory, we don't recommend using the existing Kerberos configuration file or keytab file that may already be on the domain-joined computer. See Identity Store.

kerberosKeytab
The path to the Kerberos keytab file on the local computer. It is recommended that you create a keytab file with keys specifically for Tableau Server service and that you do not share the keytab file with other applications on the computer. For example, on Linux, you might place the keytab file in the /var/opt/tableau/keytab directory.
kerberosPrincipal

The service principal name for Tableau Server on the host machine. The keytab must have permission for this principal. Do not use the existing system keytab at /etc/krb5.keytab. Rather, we recommend that you register a new service principal name. To see principals in a given keytab, run the klist -k command. See Understanding Keytab Requirements.

LDAP simple bind options

directoryservicetype

The type of directory service that you want to connect to. Either activeldirectory or openldap.

hostname

The hostname of the LDAP server. You can enter a hostname or an IP address for this value.

port

Use this option to specify the non-secure port of the LDAP server. Plaintext is usually 389.

sslPort

Use this option to specify the secure port of the LDAP server. We recommend secure LDAP for simple bind. LDAPS is usually port 636.

username

The user name that you want to use to connect to the directory service. The account that you specify must have permission to query the directory service. For Active Directory, enter the username, for example, jsmith. For LDAP servers, enter the distinguished name (DN) of the user that you want to use to connect. For example, you might enter cn=username,dc=your-local-domain,dc=lan.

password

The password of the user that you want to use to connect to the LDAP server.
Shared LDAP options

The following options can be set for generic LDAP, OpenLDAP, or Active Directory implementations.

**bind**

The way that you want to authentication communication from the Tableau Server service to the LDAP directory service. Enter `gssapi` for GSSAPI (Kerberos).

**domain**

In Windows Active Directory environments, specify the domain where Tableau Server is installed, for example, "example.lan". In LDAP directories, specify the root domain name in the same format. For example, if your root is "dc=my,dc=root", specify "my.root".

**root**

LDAP only. Do not specify for Active Directory.

If you do not use a dc component in the LDAP root or you want to specify a more complex root you need to set the LDAP root. Use the "o=my,u=root" format. For example, for the domain, example.lan, the root would be "o=example,u=lan".

**membersRetrievalPageSize**

This option determines the maximum number of results returned by an LDAP query. For example, consider a scenario where Tableau Server is importing an LDAP group that contains 50,000 users. Attempting to import such a large number of users in a single operation is not a best practice. When this option is set to 1500, Tableau Server imports the first 1500 users in the first response. After those users are processed, Tableau Server requests the next 1500 users from the LDAP server, and so forth. We recommend that you modify this option only to accommodate the requirements of your LDAP server.

**identityStoreSchemaType options**

If you configure an LDAP connection to an LDAP server, you can enter schema information specific to your LDAP server in the `identityStoreSchemaType` object. If you are con-
necting to Active Directory ("directoryServiceType": "activedirectory"), then do not configure these options.

userBaseFilter

The filter that you want to use for users of Tableau Server. For example, you might specify an object class attribute and an organization unit attribute.

userUsername

The attribute that corresponds to user names on your LDAP server.

certificate

The attribute that corresponds to user email addresses on your LDAP server.

userCertificate

The attribute that corresponds to user certificates on your LDAP server.

userThumbnail

The attribute that corresponds to user thumbnail images on your LDAP server.

userJpegPhoto

The attribute that corresponds to user profile images on your LDAP server.

groupBaseFilter

The filter that you want to use for groups of users of Tableau Server. For example, you might specify an object class attribute and an organization unit attribute.

groupName

The attribute that corresponds to group names on your LDAP server.

groupEmail
The attribute that corresponds to group email addresses on your LDAP server.

groupDescription
The attribute that corresponds to group descriptions on your LDAP server.

member
The attribute that describes the list of users in a group.

distinguishedNameAttribute
The attribute that stores the distinguished names of users. This attribute is optional, but it greatly improves the performance of LDAP queries.

serverSideSorting
Whether the LDAP server is configured for server-side sorting of query results. If your LDAP server supports server-side sorting, set this option to true. If you are unsure whether your LDAP server supports this, enter false, as misconfiguration may cause errors.

rangeRetrieval
Whether the LDAP server is configured to return a range of query results for a request. This means that groups with many users will be requested in small sets instead of all at once. LDAP servers that support range retrieval will perform better for large queries. If your LDAP server supports range retrieval, set this option to true. If you are unsure whether your LDAP server supports range retrieval, enter false, as misconfiguration may cause errors.

groupClassNames
By default Tableau Server looks for LDAP group object classes containing the string "group". If your LDAP group objects do not fit the default class name, override the default by setting this value. You can provide multiple classnames separated by commas. This option takes a list of strings, which requires passing each class in quotes, separated by a comma (no space) and within brackets. For example: [
"basegroup","othergroup"].

userClassNames
By default Tableau Server looks for LDAP user object classes containing the string “user” and “inetOrgPerson”. If your LDAP user objects do not use these default class names, override the default by setting this value. You can provide multipleclassnames separated by commas. This option takes a list of strings, which requires passing each class in quotes, separated by a comma (no space) and within brackets. For example: 

```
"userclass1",userclass2"
```

**kerberosSettings Entity**

Before you configure Kerberos authentication, review Kerberos Requirements.

Use the configuration file template below to create a json file. After you have filled in the options with the appropriate values, pass the json file and apply settings with the following commands:

```
tsm settings import -f /path/to/file.json
```

```
tsm pending-changes apply
```

The `pending-changes apply` command displays a prompt to let you know this will restart Tableau Server if the server is running. The prompt displays even if the server is stopped, but in that case there is no restart. You can suppress the prompt using the `--ignore-prompt` option, but this does not change the restart behavior. For more information, see `tsm pending-changes apply`.

**Configuration template**

Use this template to configure Kerberos settings.

All entity options are case sensitive.

For more explanation about configuration files, entities, and keys see Configuration File Example.

After you have finished with the initial configuration of Kerberos authentication, use the `tsm authentication kerberos <commands>` sub-category to set additional values.
Configuration file reference

The following list includes all of the options that can be included with the "kerberosSettings" entity set.

Option | Value
---|---

Enabled

Options: true or false.

Enables Kerberos authentication.

KeytabFile

Required.

Path to valid Kerberos keytab file.

dBClasses

Comma-separated list of database classes for global credentials. May be required for connecting to Cloudera data sources.
mutualSSLSettings Entity

Before you configure mutual SSL, review Configure SSL for External HTTP Traffic to and from Tableau Server.

The mutualSSLSettings entity combines both SSL and mutual SSL configuration. Mutual SSL requires that external SSL has been enabled and properly configured.

The TSM entities use JSON and key-value pairs. Use the configuration file template below to create a .json file. Provide values for the appropriate keys for your environment, and then pass the .json file to Tableau Server with the following commands:

```bash
tsm settings import -f <path-to-file.json>

 tsm pending-changes apply
```

The pending-changes apply command displays a prompt to let you know this will restart Tableau Server if the server is running. The prompt displays even if the server is stopped, but in that case there is no restart. You can suppress the prompt using the --ignore-prompt option, but this does not change the restart behavior. For more information, see tsm pending-changes apply.

Configuration template

Use this template to configure mutual SSL settings.

All entity options are case sensitive.

For more explanation about configuration files, entities, and keys see Configuration File Example.

```json
{
   "configEntities": {
      "mutualSSLSettings": {
         "_type": "mutualSSLSettingsType",
```
"sslEnabled": true,
"proxyLogin": false,
"clientCertRequired": true,
"caCertFile": "required",
"keyFileName": "required",
"keyPassphrase": "",
"chainFile": "",
"revocationFile": "",
"redirect": false,
"fallbackToPassword": true,
"protocols": "",
"cipherSuite": "",
"forceHttpsForPublicEmbed": false
}
}
}

Configuration file reference

sslEnabled

Enable SSL. This is a prerequisite to enabling mutual SSL.

clientCertRequired (MutualSSL)

Set to true to enable mutual SSL authentication. Set to false to disable.

ciaCertFile (MutualSSL)

Required.

Specify the CA-issued certificate file for two-way SSL. The file path must be readable by Tableau Server.
certFileName

Specify the file that contains the concatenation of PEM encoded CA certificates that form the certificate chain for the server certificate.

Alternatively the referenced file can be the same as caCertFile when the CA certificates are directly appended to the server certificate for convenience.

keyFileName

If the key is not combined with the certificate, use this configuration key to point to the key file. If you have both an RSA and a DSA private key, you can configure both in parallel (for example, to also allow the use of DSA ciphers).

keyPassphrase

Optional. Passphrase for the certificate file. The passphrase you enter will be encrypted while at rest.

**Note:** If you create a certificate key file with a passphrase, you cannot reuse the SSL certificate key for SAML.

revocationFile

Specifies the file path for an SSL CA Certificate Revocation List (.crl) file.

Redirect

Default: true. Specifies whether Tableau Server should redirect http requests as https requests to the appropriate endpoint.

clientCertMapping (MutualSSL)

Specifies the method for retrieving the user name from the certificate.

**Accepted values:** ldap, upn, cn
For a server using local authentication, the default setting is `upn` (User Principal Name).

When Tableau Server authentication is configured for Active Directory (AD), the default is `ldap` (Lightweight Directory Access Protocol). This tells the server to go to AD to validate the user, and it ignores the names inside the certificate.

You can set `cn` for either authentication type to use the CN in the Subject DN in the certificate.

For more information, see Mapping a Client Certificate to a User During Mutual Authentication.

`fallbackToPassword` (MutualSSL)

Set to true to give users the option to sign in to Tableau Server through their user name and password if mutual SSL authentication fails. Set to false to disallow this fallback option.

`protocols`

List the Transport Layer Security (TLS) protocol versions you want to allow or disallow.

Default value: "all -SSLv2 -SSLv3"

However, we recommend the using the following setting:

"all -SSLv2 -SSLv3 -TLSv1 -TLSv1.1"

For more information, see tsm security external-ssl enable. For general information, see the Apache online documentation.

`cipherSuite`

List ciphers to allow or disallow for SSL.

Default value:

See the OpenSSL ciphers page for cipher list format. Use caution when changing this option. The default values disallow ciphers that are no longer considered adequately secure.

proxyLogin

Default: false. Indicates that Tableau Server uses a proxy for SSL on sign-in only. It controls the protocol the server reports to Tableau Desktop for sign-in APIs.

forceHTTPForPublicEmbed

Default value: false. Forces the code for embedded views to use SSL.

openIDSettings Entity

Before you configure OpenID authentication, review Requirements for Using OpenID Connect.

Use the configuration file template below to create a json file. After you have filled in the options with the appropriate values, pass the json file and apply settings with the following commands:

```
tsm settings import -f path-to-file.json

 tsm pending-changes apply
```

The `pending-changes apply` command displays a prompt to let you know this will restart Tableau Server if the server is running. The prompt displays even if the server is stopped, but in that case there is no restart. You can suppress the prompt using the `--ignore-prompt` option, but this does not change the restart behavior. For more information, see tsm pending-changes apply.

Configuration template

Use this template to configure OpenID settings.

All entity options are case sensitive.
For more explanation about configuration files, entities, and keys see Configuration File Example.

After you have finished with the initial configuration of OIDC, use the tsm authentication openid <commands> sub-category to set additional values.

```json
{
    "configEntities": {
        "openIDSettings": {
            "_type": "openIDSettingsType",
            "enabled": true,
            "clientId": "required",
            "clientSecret": "required",
            "configURL": "required if staticFile value is not set",
            "staticFile": "required if configURL value is not set",
            "externalURL": "required"
        }
    }
}
```

**Configuration file reference**

The following list includes all of the options that can be included with the "openIDSettings" entity set.

_\_type_

Required.

Do not change.

enabled

Required.

Set to `true`. 
Tableau Server on Linux Administrator Guide

**clientId**

**Required.**

Specifies the provider client ID that your IdP has assigned to your application. For example, "laakjwdlnaoioloadjwha".

**clientSecret**

**Required.**

Specifies the provider client secret. This is a token that is used by Tableau to verify the authenticity of the response from the IdP. This value is a secret and should be kept securely.

For example, "fwahfkjaw72123=".

**configURL**

**Required.**

Specifies provider configuration URL. If you do not specify a configuration URL, then delete this option and specify a path and file name for staticFile instead.

**staticFile**

**Required.**

Specifies the local path to the static OIDC discovery JSON document. If you do not specify a static file, then delete this option and specify a url for configURL instead.

**externalURL**

**Required.**

The URL of your server. This is typically is the public name of your server, such as http://example.tableau.com.

**connectionTimeout**
Optional.

Specifies connection timeout span in seconds. Default is 10.

readTimeout

Optional.

Specifies read timeout span in seconds. Default is 30.

ignoreDomain

Set this to true if the following are true:

- You are using email addresses as usernames in Tableau Server
- You have provisioned users in the IdP with multiple domain names
- You want to ignore the domain name portion of the email claim from the IdP

Before you proceed, review the user names that will be used as a result of setting this option to true. User name conflicts may occur. In the case of a user name conflict, the risk of information disclosure is high. See Requirements for Using OpenID Connect.

ignoreJWK

Set this to true if your IdP does not support JWK validation. In this case, we recommend authenticating communication with your IdP using mutual TLS or another network layer security protocol. Default is false.

customScope

Specifies a custom scope user-related value that you can use to query the IdP. See Requirements for Using OpenID Connect.

idClaim

Change this value if your IdP does not use the subclaim to uniquely identify users in the ID token. The IdP claim that you specify should contain a single, unique string.
usernameClaim

Change this value to the IdP claim that your organization will use to match user names as stored in Tableau Server.

clientAuthentication

Specifies custom client authentication method for OpenID Connect.

To configure Tableau Server to use the Salesforce IdP, set this value to `client_secret_post`.

iFramedIDPEnabled

Set to `true` to allow IdP displayed in an iFrame. The IdP must disable clickjack protection to allow iFrame presentation.

samlSettings Entity

This article contains a template and reference for configuring server-wide SAML on Tableau Server, using a configuration file with keys and values for the `samlSettings` entity. This information supplements the SAML configuration steps in Configure Server-Wide SAML.

To create a SAML configuration template and apply it to Tableau Server, you complete the following steps:

1. Review the following two sections that describe the template and how it’s structured (Template categories and definitions and samlSettings configuration template).

2. Paste the JSON code shown in the template into a new text file, and save it using a `json` extension.

3. Use the SAML configuration entity reference to help you provide values where required.

4. Add optional key/value pairs specific to your environment. For example, if your SAML
certificate key file requires a passphrase, you will need to specify the password in the
`wgserver.saml.key.passphrase` parameter using the `tsm configuration set`
command.

5. Pass the configuration file to Tableau Server.

Template categories and definitions

The template uses placeholders for each key value. These placeholders are categorized as
follows:

- **Required**: Attributes with the "required" value must be replaced with valid data
  before you run the configuration command. Review the configuration file reference
  for valid values.

- **Hard-coded**: Attribute names that are prepended with an underscore (_), for
  example "_type" hold hard-coded values. Do not change these values.

- **Default values**: Attributes that are set to a value that is not "required" are default
  values. These are required attributes that you can change as appropriate for your
  environment.

- **Empty sets**: Values that are empty ("") can be passed as they are, or you can
  provide a value for your installation.

All entity options are case sensitive.

**samlSettings configuration template**

Paste this code into a text file and customize it for your environment, using the reference
below.

```json
{
    "configEntities": {
        "samlSettings": {
```
SAML configuration entity reference

The following table includes all of the options you can include with the "samlSettings" entity set.

idpMetadataFile

Required. The path and file name for the XML file generated by the IdP. The XML metadata must include the user name attribute (assertion).

If you completed the steps described in Configure Server-Wide SAML the value you enter here would be:

/var/opt/tableau/tableau_server/data/saml/<metadata-file.xml>

enabled

true | false

Required. Indicates whether SAML authentication is enabled and can be configured on the server.
The SAML return URL configured in the IdP. This is typically the external URL that Tableau Server users enter in their browser to access the server, such as http://tableau_server.

Notes

- http://localhost does not work for an external server.
- Adding a trailing slash to the URL (http://tableau-server/) is not supported.

entityId

Required. IdP entity ID value.

Typically the same as the Tableau Server return URL. The entity ID that you enter is used as a base for generating site-specific entity IDs. For example, if you enter "http://tableau-server", an individual site configured for SAML might display an entity ID something like the following:

http://tableau-server-/samlservice/public/sp/metadata?alias=48957410-9396-430a-967c-75b-db6e002a0

certFile

Enter the location and file name of the x509 certificate (.crt) file for SAML. For example:

/var/opt/tableau/tableau_server/data/saml/<file.crt>

For more information, see SAML Requirements and Configure Server-Wide SAML.

dKeyFile

Specify the location of the RSA or DSA private key (.key) file that accompanies the certificate file and is not password protected. For example:
For more information, see SAML Requirements and Configure Server-Wide SAML.

**idpDomainAttribute**

Optional. This value is used when Tableau Server includes user accounts from multiple domains. Specify the fully qualified domain name (FQDN) of the domain you want to be the default. For example `tableauserver.mycompany.com`.

If you do not provide a value for this key, the value used depends on the Tableau Server user authentication setting:

- For local authentication, the value `local` is used.
- For Active Directory, Tableau uses the FQDN from the configuration setting `wgserver.domain.default`.

To get the value for `wgserver.domain.default`, you can run the following command:

```bash
tsm configuration get --key wgserver.domain.default
```

**idpUsernameAttribute**

Required. In the IdP metadata, find the attribute that is used to specify user name values, and enter the name of that attribute. For example, `username`.

**idpEmailAttribute**

Optional. In the IdP metadata, the name of the attribute the IdP uses to specify users' email addresses.

**idpDisplayNameAttribute**

Optional. In the IdP metadata, the name of the attribute the IdP uses to specify users' display names.
idpAttributeUseDefaultDomain

true | false

Optional. The default value is true for a single domain or for multiple domains when you do not specify a value for idpDomainAttribute. The default value is false if you specify a non-default value for idpDomainAttribute.

desktopNoSAML

true | false

Allow users to use SAML authentication when they sign in from Tableau Desktop.

By default this is not set, so the effective behavior is equivalent to setting it to false. If single sign-on from Tableau client applications does not work with your IdP, you can set this to true to disable SAML authentication through Tableau Desktop.

appNoSAML

true | false

Allow using SAML to sign in from older versions of Tableau Mobile app. Devices running Tableau Mobile app version 19.225.1731 and higher ignore this option. To disable devices running Tableau Mobile app version 19.225.1731 and higher, disable SAML as a client login option on Tableau Server.

logoutEnabled

true | false

Specifies whether the logout UI option is displayed for users who have logged on with SAML. The default is true. This setting applies only if SAML authentication is enabled for the server.

logoutUrl

Enter the URL to redirect to after users sign out of the server. By default this is the
Tableau Server on Linux Administrator Guide

Tableau Server sign-in page. You can specify an absolute or a relative URL.

maxAuthenticationAge

Specifies the maximum number of seconds allowed between a user’s authentication and processing of the AuthNResponse message. The default value is 7200 (2 hours).

maxAssertionTime

Specifies the maximum number of seconds, from creation, that a SAML assertion is usable. Default value is 3000 (50 minutes).

sha256Enabled

true | false

Indicates whether the SAML certificate is an SHA-2 (256 or 512 bit) certificate. Default value is false. If you use the same certificate for SAML as you do for SSL, you will most likely set this to true.

signRequests

true | false

Default value is true.

iFramedIdpEnabled

true | false

Default value is false, meaning that when users select the sign-in button on an embedded view, the IdP’s sign-in form opens in a pop-up window.

When you set it to true, and a server SAML user who is already signed in navigates to a web page with an embedded view, the user will not need to sign in to see the view.

You can set this to true only if the IdP supports signing within an iframe. The iframe option is less secure than using a pop-up, so not all IdPs support it. If the IdP sign-in
page implements clickjack protection, as most do, the sign-in page cannot display in an iframe, and the user cannot sign in.

If your IdP does support signing in via an iframe, you might need to enable it explicitly. However, even if you can use this option, it disables Tableau Server clickjack protection for SAML, so it still presents a security risk.

**port**

The port that handles SAML authentication traffic.

Default value: 80

If your server is configured for SSL (recommended), set this to 443.

**singleLogoutSupported**

true | false

Enable or disable SAML single logout (SLO). This value is not set by default, so effectively it behaves as if the value were set to false. To set it to true, the IdP must support SLO.

**siteSpecific**

enabled | disabled

Specifies whether the server is configured to support site-specific SAML authentication, in which you can configure sites to use different IdPs or IdP applications. Set to disabled by default. If you set to enabled, you then configure a site’s SAML settings in the Tableau Server web UI. Sign in to the site on which you want to use SAML, and go to the Authentication tab on the Settings page.

**Pass the configuration file to Tableau Server**

After you have provided an appropriate value for each entity you include in the configuration template, use the following commands to pass the .json file and apply settings to Tableau Server.
tsm settings import -f path-to-file.json

tsm pending-changes apply

See also

After you complete the initial SAML configuration, use tsm authentication mutual-ssl <commands> to set additional values.

For the command-line reference for configuring SAML, see tsm authentication saml <commands>.

sapHanaSettings Entity

Use the sapHanaSettings entity to configure SAML delegation for SAP HANA.

Review Configure SAP HANA SSO before you continue.

Use the configuration file template below to create a json file. After you have filled in the options with the appropriate values, pass the json file and apply settings with the following commands:

tsm settings import -f path-to-file.json

tsm pending-changes apply

The pending-changes apply command displays a prompt to let you know this will restart Tableau Server if the server is running. The prompt displays even if the server is stopped, but in that case there is no restart. You can suppress the prompt using the --ignore-prompt option, but this does not change the restart behavior. For more information, see tsm pending-changes apply.
SAP HANA SAML settings

The settings in the template below specify the settings for Tableau Server in a SAML delegation scenario with SAP HANA.

Configuration template

Use this template to configure trusted authentication settings.

All that are referenced in configEntities must be located on the local computer. Do not specify UNC paths.

All entity options are case sensitive.

For more explanation about configuration files, entities, and keys see Configuration File Example.

```json
{
  "configEntities": {
    "sapHanaSettings": {
      "_type": "sapHanaSettingsType",
      "enabled": "true",
      "usernameFormat": "username",
      "usernameCase": "preserve",
      "certFile": "path-to-cert_file",
      "keyFile": "path-to-key_file"
    }
  }
}
```

Configuration file reference

This table includes all of the options that can be included with the "gatewaySettings" entity set.

enabled

Required.
Tableau Server on Linux Administrator Guide

**Values:** true or false

usernameFormat

**Values:** username, domain_and_username, or email

Specifies user name credential format.

usernameCase

**Values:** lower, upper, or preserve

Specifies user input name case.

certFile

Specifies file path and name for the certificate file on the local computer.

For example, 
"/var/opt/tableau/tableau_server/data/saml/saml.cert.crt".

cKeyFile

Specifies file path and name for the certificate key on the local computer.

For example, 
"/var/opt/tableau/tableau_server/data/saml/saml.key.der".

**shareProductUsageDataSettings Entity**

Before you configure this entity, see Product usage data to better understand the behavioral and usage data that Tableau collects. (Your confidential database values are never included.)

Use the configuration file template below to create a json file. After you have filled in the options with the appropriate values, pass the json file and apply settings with the following commands:


```bash
  tsm settings import -f /path/to/file.json

  tsm pending-changes apply
```

The `pending-changes apply` command displays a prompt to let you know this will restart Tableau Server if the server is running. The prompt displays even if the server is stopped, but in that case there is no restart. You can suppress the prompt using the `--ignore-prompt` option, but this does not change the restart behavior. For more information, see `tsm pending-changes apply`.

### Configuration template

Use this template to configure product usage data settings.

All entity options are case sensitive.

For more explanation about configuration files, entities, and keys see Configuration File Example.

```json
{
  "configEntities": {
    "shareProductUsageDataSettings": {
      "_type": "shareProductUsageDataSettingsType",
      "enabled": "true",
    }
  }
}
```

### Configuration file reference

The following list includes all of the options that can be included with the "shareProductUsageDataSettings" entity set:

- `_type`

  Required value: `shareProductUsageDataSettingsType`
enabled

Options: true or false

The default, true, shares product usage data from your server with Tableau.

trustedAuthenticationSettings Entity

Before you configure trusted authentication, review Trusted Authentication.

Use the configuration file template below to create a json file. After you have filled in the options with the appropriate values, pass the json file and apply settings with the following commands:

tsm settings import -f /path/to/file.json

 tsm pending-changes apply

The pending-changes apply command displays a prompt to let you know this will restart Tableau Server if the server is running. The prompt displays even if the server is stopped, but in that case there is no restart. You can suppress the prompt using the --ignore-prompt option, but this does not change the restart behavior. For more information, see tsm pending-changes apply.

Configuration template

Use this template to configure trusted authentication settings.

All entity options are case sensitive.

For more explanation about configuration files, entities, and keys see Configuration File Example.

After you have finished with the initial configuration of trusted authentication, use the tsm authentication trusted <commands> sub-category to set additional values.

{
  "configEntities": {
}
"trustedAuthenticationSettings": {
    "_type": "trustedAuthenticationSettingsType",
    "trustedHosts": ["webserv1","webserv2","webserv3"]
}

Configuration file reference

The following list includes all of the options that can be included with the "trustedAuthenticationSettings" entity set.

trustedHosts

Required.

IP address or host names of web servers that request trusted tickets from Tableau Server.

This option takes a list of strings, which requires passing each IP or host in quotes, separated by a comma (no space) and within brackets. For example: ["192.168.1.101","192.168.1.102","192.168.1.103"] or ["web-serv1","webserv2","webserv3"].

The values you specify overwrite previous settings. Therefore, you must include the full list of hosts when you configure this value.

tokenLength

Optional.

The value can be set to any integer between 9 and 255, inclusive.

Determines the number of characters in each trusted ticket. The default setting of 24 characters provides 144 bits of randomness. This option is ignored unless useV2Tickets is set to true, which is not a recommended best practice.

logLevel
Optional.

```
all | debug | info | warn | error | fatal | off
```

**Default:** info

Specifies logging level for processes related to creating and redeeming trusted tickets. See Change Logging Levels.

**timeoutInSeconds**

Optional.

**Default:** 180

Specifies the length of time (in seconds) to invalidate trusted tickets after they are created.

**tryCount**

Optional.

**Default:** 10

Specifies the number of times to attempt to create a trusted ticket entry.

**use9DigitToken**

Optional.

```
true | false
```

**Default:** false

When set to `true`, tickets are 9 digits long (as in version 8.0 and earlier) and the option `tokenLength` is ignored. This option is intended for temporary support of legacy code.
**useV2Tickets**

Optional.

`true` | `false`

Default: `false`

Specifies whether Tableau Server should return a legacy URL format for trusted ticket requests. The legacy URL format includes a 24 character, Base64-encoded string. Beginning with Tableau Server 10.3, the URL that is returned has been updated and includes a Base64-encoded UUID and a 24 character secure random string. Only set option this to `true` if you have deployed trusted tickets with custom code that requires the legacy URL format. We recommend instead, updating your custom code to accept the new URL format.

---

**web-data-connector-settings Entity**

This entity is used to manage web data connector (WDC) settings. To learn more about using WDCs in Tableau Server, see Web Data Connectors in Tableau Server and tsm data-access.

Use the configuration file template below to create a `.json` file. After you have filled in the options with the appropriate values, pass the `.json` file and apply settings with the following commands:

```
  tsm settings import -f /path/to/file.json
  tsm pending-changes apply
```

The `pending-changes apply` command displays a prompt to let you know this will restart Tableau Server if the server is running. The prompt displays even if the server is
stopped, but in that case there is no restart. You can suppress the prompt using the --ignore-prompt option, but this does not change the restart behavior. For more information, see tsm pending-changes apply.

Web data connector settings

The web data connector (WDC) settings in the template below specify whether WDCs are enabled, whether refresh of WDCs is enabled, and the primary and secondary safe lists. The safe lists (or “whitelists”) indicate which WDC URLs are approved for use in your Tableau Server installation, and the domains or URLs that a connector can send requests to and receive requests from.

Configuration template

Use this template to configure the WDC settings.

All entity options are case sensitive.

For more explanation about configuration files, entities, and keys see Configuration File Example.

```json
{
  "configEntities": {
    "web-data-connector-settings": {
      "_type": "webDataConnectorSettingsType",
      "refreshEnabled": true,
      "whitelist": {
        "https://dtreskunov.github.io:80/wta-wdc/": {
          "secondaryWhitelist": [
            "(https://dtreskunov.github.io/*)"
          ]
        }
      },
      "enabled": true
    }
  }
}
```
Configuration file reference

This table includes all of the options that can be included with the "web-data-connector-settings" entity set.

_type

Required.

Value: "web-data-connector-settings"

Do not change.

refreshEnabled

Set to false to disable refresh of WDCs. Defaults to true.

whitelist

Required.

Can contain one or more matching sets of safe lists and secondary safe lists (one set per WDC). The first URL provided is the safe list, where you specify the WDC URL and port, formatted as follows:

<scheme>://<host>:<port>/<path>

For many WDCs, the <port> value is 443 or 80, but you can check the value for your connector by looking at the data source details on Tableau Server or Tableau Online.

secondaryWhitelist

Required.
Tableau Server on Linux Administrator Guide

Specifies the domains or URLs that a connector can send requests to and receive requests from. To specify a domain, use a wildcard character (*), as in the following example:

https://dtreskunov.github.io/*

enabled

Set to false to disable use of WDCs. Defaults to true.

tabcmd

Tableau provides the tabcmd command-line utility which you can use to automate site administration tasks on your Tableau Server site. For example, creating or deleting users, projects, and groups.

This utility is included with Tableau Server, and is automatically installed on the initial server node. You can also run it from other computers, even computers that are not part of your Tableau Server installation, but to do so you need to download the tabcmd installer from the Tableau website. For more information, see Install tabcmd below.

Install tabcmd

When Tableau Server or Tableau Online is upgraded to a new version, if an updated version of tabcmd is required, you can download it from the Tableau Releases page on the Tableau website, to avoid issues caused by version incompatibilities. Using an out of date version of tabcmd can cause errors and unpredictable results.

1. Open a web browser and go to the Tableau Server Releases page (do this even if you use Tableau Online).

2. For Tableau Online: select the latest release.

For Tableau Server (Windows or Linux): select the release that matches your server version.
If the expanded information shows maintenance releases, select the latest maintenance release or the one that matches your server version.

This takes you to the release notes page, where you can read about security improvements and resolved issues.

3. Scroll to the **Download Files** section under the resolved issues, select the `tabcmd` download link that is compatible with the computer on which you’ll run the `tabcmd` commands.

The remaining steps refer to this computer as “the `tabcmd` computer.”
4. Save the installer to the tabcmd computer, or a location accessible from that computer (a mounted drive, for example).

5. Complete the installation steps as appropriate for the operating system of the tabcmd computer:

- **Windows**

  By default tabcmd is installed to `C:\Program Files\Tableau\Tableau Server\<version>\extras\Command Line Utility`. You can change this during installation and recommend that you install tabcmd to a folder named `tabcmd` at the root of the C:\ drive (`C:\tabcmd`). This can make it easier to locate and run, and will accommodate some limitations with the Windows operating system if you add the tabcmd directory to the Windows PATH.

  **Note** The tabcmd Setup program does not add the tabcmd directory to the Windows PATH variable. You can add it manually, or you can include the full path to tabcmd each time you call it.

  You can install tabcmd in two ways on Windows:
Double-click the installer to follow the steps in the UI:

a. Accept the license agreement.

b. If you want to install to a non-default location, click **Customize** and type or browse to the location you want to install tabcmd to.

c. Click **Install**.

If you are prompted by Windows Defender Firewall or User Account Control, click **Allow access**.

Run the installer from a command prompt:

a. Open a command prompt as administrator on the tabcmd computer.

b. Navigate to the directory where you copied the tabcmd installer.

c. Install tabcmd:

```
tableau-setup-tabcmd-tableau-<version_code>-x64.exe /quiet ACCEPTEULA=1
```

To install to a non-default location:

```
tableau-setup-tabcmd-tableau-<version_code>-x64.exe /quiet ACCEPTEULA=1 INSTALLDIR=R="<path\to\install\directory>"
```

For example:

```
tableau-setup-tabcmd-tableau-<version_code>-x64.exe /quiet ACCEPTEULA=1 INSTALLDIR=R="C:\tabcmd"
```
Tableau Server on Linux Administrator Guide

For a complete list of command line options you can use with the tabcmd installer, run the installer with a /?. For more information on tabcmd installer command line options, see Install Switches and Properties for tabcmd.

The tabcmd Setup program creates logs in C:\Users\<user>\AppData\Local\Temp you can use if you have problems installing tabcmd. The logs use the naming convention Tableau_Server_Command_Line.Utility_(<version_code>)_#####.log.

• Linux

**Note:** To run tabcmd on a Linux computer, you must have Java 1.8 installed. On RHEL-like systems this will be installed as a dependency when you install tabcmd. On Debian-like systems, you need to install Java 1.8 separately if it is not already installed.

a. Log on as a user with sudo access to the tabcmd computer.

b. Navigate to the directory where you copied the .rpm or .deb package that you downloaded.

• On RHEL-like distributions, including CentOS, run the following command:

```
sudo yum install tableau-tabcmd-<version>.noarch.rpm
```

• On Ubuntu and Debian, run the following command:
How to use tabcmd

The basic steps for using tabcmd are as follows:

1. Open the Command Prompt as an administrator.

2. On a Windows computer, if you installed tabcmd on a computer other than the initial node, change to the directory where you installed tabcmd.

   On a Linux computer, you do not need to change to the install directory.

3. Run the tabcmd command.

When you use tabcmd, you must establish an authenticated server session. The session identifies the server or Tableau Online site and the user running the session. You can start a session first, and then specify your command next, or you can start a session and execute a command all at once.

**Important:** If you are using tabcmd to perform more than one task, you must run tasks one after another (serially), rather than at the same time (in parallel).

Commands (such as `login`) and the options (such as `-s`, `-u`, etc.) are not case sensitive, but the values you provide (such as `User@Example.com`) are case sensitive.

Examples

The following command demonstrates starting a session with the Tableau Server named `tabserver.mycompany.com`:

```
tabcmd login -s http://tabserver.mycompany.com -u admin -p mypassword
```
The next example shows a command that deletes a workbook named *Sales_Workbook*:

```
tabcmd delete "Sales_Workbook"
```

Here's how to accomplish all of the above with one command—note that you do not need login here:

```
tabcmd delete "Sales_Workbook" -s http://tabserver.mycompany.com -u admin -p mypassword
```

A Tableau Server can run multiple sites. When a workbook is on the Default site of a multi-site server you don’t need to specify Default, the above command is sufficient. However, if the command applies to something on a site other than Default, you need to specify the site ID for that site (see login). Here's the same command for a workbook that's on the West Coast Sales site (site ID `wsales`):

```
tabcmd delete "Sales_Workbook" -s http://tabserver.mycompany.com -t wsales -u admin -p mypassword
```

The options `-s`, `-t`, `-u`, and `-p` are among the `tabcmd` global variables, which can be used with any command.

For more information, see `tabcmd Commands`.

### `tabcmd` Commands

You can use the following commands with the `tabcmd` command line tool:

- `addusers` (to group)
- `createextracts`
- `creategroup`
- `createproject`
- `createsite`
- `createsiteusers`
- `createusers`
decryptextracts
delete workbook-name or datasource-name
deleteextracts
deletegroup
deleteproject
deletesite
deletesiteusers
deleteusers
editedomain
editsite
encryptextracts
export
get url
initialuser
listdomains
listsites
login
logout
publish
publishsamples
reencryptextracts
refreshextracts
removeusers
reset openid_sub
runschedule
set
syncgroup
version

addusers group-name

Adds users to the specified group.

Example

tabcmd addusers "Development" --users "users.csv"
Tableau Server on Linux Administrator Guide

Options

--users

Add the users in the given .csv file to the specified group. The file should be a simple list with one user name per line. User names are not case sensitive. The users should already be created on Tableau Server.

For more information, see CSV Import File Guidelines.

--[no-]complete

When set to complete this option requires that all rows be valid for any change to succeed. If not specified, --complete is used.

Global options

The following options are used by all tabcmd commands. The --server, --user, and --password options are required at least once to begin a session. An authentication token is stored so subsequent commands can be run without including these options. This token remains valid for five minutes after the last command that used it.

-h, --help

Displays the help for the command.

-c, --use-certificate

Use client certificate to sign in. Required when mutual SSL is enabled.

For information about configuring the certificate, start with the following topic appropriate for your Tableau Server OS:
**Windows:** Configure Mutual SSL

**Linux:** Configure Mutual SSL

- `-s`, `--server`

  The Tableau Server URL, which is required at least once to begin session.

- `-u`, `--user`

  The Tableau Server username, which is required at least once to begin session.

- `-p`, `--password`

  The Tableau Server password, which is required at least once to begin session.

- `--password-file`

  Allows the password to be stored in the given `.txt` file rather than the command line for increased security.

- `-t`, `--site`

  Indicates that the command applies to the site specified by the Tableau Server site ID. If you do not specify a site, the Default site is assumed. Applies only to servers with multiple sites.

- `-x`, `--proxy`

  Host:Port

  Uses the specified HTTP proxy.

- `--no-prompt`

  When specified, the command will not prompt for a password. If no valid password is
Tableau Server on Linux Administrator Guide

provided the command will fail.

--no-proxy

When specified, an HTTP proxy will not be used.

--no-certcheck

When specified, tabcmd (the client) does not validate the server’s SSL certificate.

--[no-]cookie

When specified, the session ID is saved on login so subsequent commands will not need to log in. Use the no- prefix to not save the session ID. By default the session is saved.

--timeout

Waits the specified number of seconds for the server to complete processing the command. By default the process will timeout in 30 seconds.

--

Specifies the end of options on the command line. You can use -- to indicate to tabcmd that anything that follows -- should not be interpreted as an option setting and can instead be interpreted as a value for the command. This is useful if you need to specify a value in the command that includes a hyphen. The following example shows how you might use -- in a tabcmd command, where -430105/SHEET1 is a required value for the export command.

tabcmd export --csv -f "D:\export10.csv" -- -430105/SHEET1

createextracts

Creates extracts for a published workbook or data source.
Options

-d, --datasource

The name of the target data source for extract creation.

--embedded-datasources

A space-separated list of embedded data source names within the target workbook. Enclose data source names with double quotes if they contain spaces. Only available when creating extracts for a workbook.

--encrypt

Create encrypted extract.

--include-all

Include all embedded data sources within target workbook. Only available when creating extracts for workbook.

--parent-project-path

Path of the project that is the parent of the project that contains the target resource. Must specify the project name with --project.

--project

The name of the project that contains the target resource. Only necessary if --workbook or --datasource is specified. If unspecified, the default project 'Default' is used.

-u, -url

The canonical name for the resource as it appears in the URL.

-w, -workbook
The name of the target workbook for extract creation.

Global options

The following options are used by all `tabcmd` commands. The `--server`, `--user`, and `--password` options are required at least once to begin a session. An authentication token is stored so subsequent commands can be run without including these options. This token remains valid for five minutes after the last command that used it.

-`h`, `--help`

Displays the help for the command.

-`c`, `--use-certificate`

Use client certificate to sign in. Required when mutual SSL is enabled.

For information about configuring the certificate, start with the following topic appropriate for your Tableau Server OS:

- **Windows**: Configure Mutual SSL
- **Linux**: Configure Mutual SSL

-`s`, `--server`

The Tableau Server URL, which is required at least once to begin session.

-`u`, `--user`

The Tableau Server username, which is required at least once to begin session.

-`p`, `--password`

The Tableau Server password, which is required at least once to begin session.
--password-file

Allows the password to be stored in the given .txt file rather than the command line for increased security.

-t, --site

Indicates that the command applies to the site specified by the Tableau Server site ID. If you do not specify a site, the Default site is assumed. Applies only to servers with multiple sites.

-x, --proxy

Host:Port

Uses the specified HTTP proxy.

--no-prompt

When specified, the command will not prompt for a password. If no valid password is provided the command will fail.

--no-proxy

When specified, an HTTP proxy will not be used.

--no-certcheck

When specified, tabcmd (the client) does not validate the server's SSL certificate.

--[no-]cookie

When specified, the session ID is saved on login so subsequent commands will not need to log in. Use the no- prefix to not save the session ID. By default the session is saved.
--timeout

Waits the specified number of seconds for the server to complete processing the command. By default the process will timeout in 30 seconds.

--

Specifies the end of options on the command line. You can use -- to indicate to tabcmd that anything that follows -- should not be interpreted as an option setting and can instead be interpreted as a value for the command. This is useful if you need to specify a value in the command that includes a hyphen. The following example shows how you might use -- in a tabcmd command, where -430105/Sheet1 is a required value for the export command.

tabcmd export --csv -f "D:\export10.csv" -- -430105/Sheet1

creategroup group-name

Creates a group. Use addusers (for local groups) and syncgroup (for Active Directory groups) commands to add users after the group has been created.

Example

tabcmd creategroup "Development"

Global options

The following options are used by all tabcmd commands. The --server, --user, and --password options are required at least once to begin a session. An authentication token is stored so subsequent commands can be run without including these options. This token remains valid for five minutes after the last command that used it.

-h, --help

Displays the help for the command.
-c, --use-certificate

Use client certificate to sign in. Required when mutual SSL is enabled.

For information about configuring the certificate, start with the following topic appropriate for your Tableau Server OS:

- **Windows**: Configure Mutual SSL
- **Linux**: Configure Mutual SSL

-s, --server

The Tableau Server URL, which is required at least once to begin session.

-u, --user

The Tableau Server username, which is required at least once to begin session.

-p, --password

The Tableau Server password, which is required at least once to begin session.

--password-file

Allows the password to be stored in the given .txt file rather than the command line for increased security.

-t, --site

Indicates that the command applies to the site specified by the Tableau Server site ID. If you do not specify a site, the Default site is assumed. Applies only to servers with multiple sites.

-x, --proxy

Host:Port
Tableau Server on Linux Administrator Guide

Uses the specified HTTP proxy.

--no-prompt

When specified, the command will not prompt for a password. If no valid password is provided the command will fail.

--no-proxy

When specified, an HTTP proxy will not be used.

--no-certcheck

When specified, tabcmd (the client) does not validate the server’s SSL certificate.

--[no-]cookie

When specified, the session ID is saved on login so subsequent commands will not need to log in. Use the no- prefix to not save the session ID. By default the session is saved.

--timeout

Waits the specified number of seconds for the server to complete processing the command. By default the process will timeout in 30 seconds.

--

Specifies the end of options on the command line. You can use -- to indicate to tabcmd that anything that follows -- should not be interpreted as an option setting and can instead be interpreted as a value for the command. This is useful if you need to specify a value in the command that includes a hyphen. The following example shows how you might use -- in a tabcmd command, where -430105/Sheet1 is a required value for the export command.

tabcmd export --csv -f "D:\export10.csv" -- -430105/Sheet1
createproject *project-name*

Creates a project.

**Example**

```
tabcmd createproject -n "Quarterly_Reports" -d "Workbooks showing quarterly sales reports."
```

**Options**

- `-n`, `--name`

  Specifies the name of the project that you want to create.

- `--parent-project-path`

  Specifies the name of the parent project for the nested project as specified with the `-n` option. For example, to specify a project called "Nested" that exists in a "Main" project, use the following syntax: `--parent-project-path "Main" -n "Nested"`.

- `-d`, `--description`

  Specifies a description for the project.

**Global options**

The following options are used by all `tabcmd` commands. The `--server`, `--user`, and `--password` options are required at least once to begin a session. An authentication token is stored so subsequent commands can be run without including these options. This token remains valid for five minutes after the last command that used it.

- `-h`, `--help`

  Displays the help for the command.
-c, --use-certificate

Use client certificate to sign in. Required when mutual SSL is enabled.

For information about configuring the certificate, start with the following topic appropriate for your Tableau Server OS:

- **Windows**: Configure Mutual SSL
- **Linux**: Configure Mutual SSL

-s, --server

The Tableau Server URL, which is required at least once to begin session.

-u, --user

The Tableau Server username, which is required at least once to begin session.

-p, --password

The Tableau Server password, which is required at least once to begin session.

--password-file

Allows the password to be stored in the given .txt file rather than the command line for increased security.

-t, --site

Indicates that the command applies to the site specified by the Tableau Server site ID. If you do not specify a site, the Default site is assumed. Applies only to servers with multiple sites.

-x, --proxy

Host:Port
Uses the specified HTTP proxy.

--no-prompt

When specified, the command will not prompt for a password. If no valid password is provided the command will fail.

--no-proxy

When specified, an HTTP proxy will not be used.

--no-certcheck

When specified, tabcmd (the client) does not validate the server's SSL certificate.

--[no-]cookie

When specified, the session ID is saved on login so subsequent commands will not need to log in. Use the no- prefix to not save the session ID. By default the session is saved.

--timeout

Waits the specified number of seconds for the server to complete processing the command. By default the process will timeout in 30 seconds.

--

Specifies the end of options on the command line. You can use -- to indicate to tabcmd that anything that follows -- should not be interpreted as an option setting and can instead be interpreted as a value for the command. This is useful if you need to specify a value in the command that includes a hyphen. The following example shows how you might use -- in a tabcmd command, where -430105/Sheet1 is a required value for the export command.

tabcmd export --csv -f "D:\export10.csv" -- -430105/Sheet1
createsite site-name

Creates a site.

Examples

Create a site named West Coast Sales. A site ID of WestCoastSales will be automatically created, the site will have no storage quota limit, and site administrators will be able to add and remove users:

```
tabcmd createsite "West Coast Sales"
```

Create a site named West Coast Sales with a site ID of wsales:

```
tabcmd createsite "West Coast Sales" -r "wsales"
```

Prevent site administrators from adding users to the site:

```
tabcmd createsite "West Coast Sales" --no-site-mode
```

Set a storage quota, in MB:

```
tabcmd createsite "West Coast Sales" --storage-quota 100
```

Options

- -r, --url

    Used in URLs to specify the site. Different from the site name.

- --user-quota

    Maximum number of users that can be added to the site.

- --[no-]site-mode

    Allows or denies site administrators the ability to add users to or remove users from the
--storage-quota

In MB, the amount of workbooks, extracts, and data sources that can be stored on the site.

--extract-encryption-mode

The extract encryption mode for the site can be enforced, enabled or disabled. For more information, see Extract Encryption at Rest.

Global options

The following options are used by all `tabcmd` commands. The `--server`, `--user`, and `--password` options are required at least once to begin a session. An authentication token is stored so subsequent commands can be run without including these options. This token remains valid for five minutes after the last command that used it.

-h, --help

Displays the help for the command.

-c, --use-certificate

Use client certificate to sign in. Required when mutual SSL is enabled.

For information about configuring the certificate, start with the following topic appropriate for your Tableau Server OS:

- **Windows**: Configure Mutual SSL
- **Linux**: Configure Mutual SSL
Tableau Server on Linux Administrator Guide

-s, --server

The Tableau Server URL, which is required at least once to begin session.

-u, --user

The Tableau Server username, which is required at least once to begin session.

-p, --password

The Tableau Server password, which is required at least once to begin session.

--password-file

Allows the password to be stored in the given .txt file rather than the command line for increased security.

-t, --site

Indicates that the command applies to the site specified by the Tableau Server site ID. If you do not specify a site, the Default site is assumed. Applies only to servers with multiple sites.

-x, --proxy

Host:Port

Uses the specified HTTP proxy.

--no-prompt

When specified, the command will not prompt for a password. If no valid password is provided the command will fail.

--no-proxy
When specified, an HTTP proxy will not be used.

--no-certcheck

When specified, tabcmd (the client) does not validate the server’s SSL certificate.

--[no-]cookie

When specified, the session ID is saved on login so subsequent commands will not need to log in. Use the no- prefix to not save the session ID. By default the session is saved.

--timeout

Waits the specified number of seconds for the server to complete processing the command. By default the process will timeout in 30 seconds.

--

Specifies the end of options on the command line. You can use -- to indicate to tabcmd that anything that follows -- should not be interpreted as an option setting and can instead be interpreted as a value for the command. This is useful if you need to specify a value in the command that includes a hyphen. The following example shows how you might use -- in a tabcmd command, where -430105/SHEET1 is a required value for the export command.

    tabcmd export --csv -f "D:\export10.csv" -- -430105/SHEET1

create site users filename.csv

Adds users to a site, based on information supplied in a comma-separated values (CSV) file. If the user is not already created on the server, the command creates the user before adding that user to the site.
The CSV file must contain one or more user names and can also include (for each user) a password, full name, license type, administrator level, publisher (yes/no), and email address. For information about the format of the CSV file, see CSV Import File Guidelines.

As an alternative to including administrator level and publisher permissions in the CSV file, you can pass access level information by including the --role option and specifying the site role you want to assign users listed in the CSV file.

By default, users are added to the site that you are logged in to. To add users to a different site, include the global --site option and specify that site. (You must have permissions to create users on the site you specify.)

If the server contains multiple sites, you cannot add server (system) administrators through the createsiteusers command. Use createusers instead. If you specify the Server-Administrator site role for the --role option, the command returns an error. If the CSV file includes System as value for administrator, the value is ignored and the user is assigned the Unlicensed license type.

If the server contains only one site (the default site), you can specify system for the administrator value for a user, or even assign the ServerAdministrator site role using the --role option, if you want all users in the CSV file to be server administrators.

By default, this command creates users using a synchronous operation (it waits for all operations to complete before proceeding). You can use the --no-wait option to specify an asynchronous operation.

Local authentication

If the server is configured to use local authentication, the information in the CSV file is used to create users.

Active Directory authentication

If the server is configured to use Active Directory authentication, user information is imported from Active Directory, and password and friendly name information in the CSV file is ignored. Further, if a user is specified in the CSV file but no corresponding user exists in
Active Directory, the user is not added to Tableau Server. For Active Directory users, because the user name is not guaranteed to be unique across domains, you must include the domain as part of the user name. You can specify this as either domain/username or username@domain.com; however, we recommend using the domain/username format. For more information, see User Management in Active Directory Deployments.

Example

tabcmd createsiteusers "users.csv" --role "Explorer"

Options

--admin-type

Deprecated. Use the --role option instead.

--complete

Requires that all rows be valid for any change to succeed. This is the default setting.

--no-complete

Specifies that the command should make changes on the server even if not all rows contain valid information. Rows that contain invalid information are skipped.

--no-publisher

Deprecated. Use the --role option instead.

--nowait

Do not wait for asynchronous jobs to complete.

--publisher

Deprecated. Use the --role option instead.
--role

Specifies a site role for all users in the .csv file. When you want to assign site roles using the --role option, create a separate CSV file for each site role.

Valid values are ServerAdministrator, SiteAdministratorCreator, SiteAdministratorExplorer, Creator, ExplorerCanPublish, Explorer, Viewer, ReadOnly, and Unlicensed.

The default is Unlicensed for new users and unchanged for existing users. Users are added as unlicensed also if you have a user-based server installation, and if the createsiteusers command creates a new user, but you have already reached the limit on the number of licenses for your users.

Note: On a multi-site Tableau Server, if you want to assign the ServerAdministrator site role using the --role option, use the createusers command instead of createsiteusers.

--silent-progress

Do not display progress messages for the command.

Global options

The following options are used by all tabcmd commands. The --server, --user, and --password options are required at least once to begin a session. An authentication token is stored so subsequent commands can be run without including these options. This token remains valid for five minutes after the last command that used it.

-h, --help

Displays the help for the command.
-c, --use-certificate

Use client certificate to sign in. Required when mutual SSL is enabled.

For information about configuring the certificate, start with the following topic appropriate for your Tableau Server OS:

- Windows: Configure Mutual SSL
- Linux: Configure Mutual SSL

-s, --server

The Tableau Server URL, which is required at least once to begin session.

-u, --user

The Tableau Server username, which is required at least once to begin session.

-p, --password

The Tableau Server password, which is required at least once to begin session.

--password-file

Allows the password to be stored in the given .txt file rather than the command line for increased security.

-t, --site

Indicates that the command applies to the site specified by the Tableau Server site ID. If you do not specify a site, the Default site is assumed. Applies only to servers with multiple sites.

-x, --proxy

Host:Port
Tableau Server on Linux Administrator Guide

Uses the specified HTTP proxy.

```
--no-prompt
```

When specified, the command will not prompt for a password. If no valid password is provided the command will fail.

```
--no-proxy
```

When specified, an HTTP proxy will not be used.

```
--no-certcheck
```

When specified, tabcmd (the client) does not validate the server’s SSL certificate.

```
--[no-]cookie
```

When specified, the session ID is saved on login so subsequent commands will not need to log in. Use the no- prefix to not save the session ID. By default the session is saved.

```
--timeout
```

Waits the specified number of seconds for the server to complete processing the command. By default the process will timeout in 30 seconds.

```
--
```

Specifies the end of options on the command line. You can use -- to indicate to tabcmd that anything that follows -- should not be interpreted as an option setting and can instead be interpreted as a value for the command. This is useful if you need to specify a value in the command that includes a hyphen. The following example shows how you might use -- in a tabcmd command, where -430105/SHEET1 is a required value for the export command.

```
tabcmd export --csv -f "D:\export10.csv" -- -430105/SHEET1
```

createusers filename.csv

Create users in Tableau Server, based on information supplied in a comma-separated values (CSV) file.

The CSV file must contain one or more user names and can also include (for each user) a password, full name, license type, administrator level, publisher (yes/no), and email address. For information about the format of the CSV file, see CSV Import File Guidelines.

As an alternative to including administrator level and publisher permissions in the CSV file, you can pass access level information by including the --role option and specifying the site role you want to assign users listed in the CSV file.

If the server has only one site (the default site), the user is created and added to the site. If the server has multiple sites, the user is created but is not added to any site. To add users to a site, use createsiteusers.

If you have a user-based server installation, and if the command creates a new user but you have already reached the limit on the number of licenses for your users, the user is added as an unlicensed user.

Local authentication

If the server is configured to use local authentication, the information in the CSV file is used to create users.

Active Directory authentication

If the server is configured to use Active Directory authentication, user information is imported from Active Directory, and password and friendly name information in the CSV file is ignored. Further, if a user is specified in the CSV file but no corresponding user exists in Active Directory, the user is not added to Tableau Server. For Active Directory users, because the user name is not guaranteed to be unique across domains, you must include the domain as part of the user name. You can specify this as either domain/username or user-
name@domain.com; however, we recommend using the domain/username format. For more information, see User Management in Active Directory Deployments.

Example

tabcmd createusers "users.csv" --role "ServerAdministrator"

tabcmd createusers "users.csv"

Options

--admin-type

Deprecated. Use the --role option instead.

--complete

Requires that all rows be valid for any change to succeed. This is the default setting.

--no-complete

Specifies that the command should make changes on the server even if not all rows contain valid information. Rows that contain invalid information are skipped.

--no-publisher

Deprecated. Use the --role option instead.

--nowait

Do not wait for asynchronous jobs to complete.

--publisher

Deprecated. Use the --role option instead.
-r, --role

Specifies a site role for all users in the .csv file. When you want to assign site roles using the --role option, create a separate CSV file for each site role.

Valid values are ServerAdministrator, SiteAdministratorCreator, SiteAdministratorExplorer, Creator, ExplorerCanPublish, Explorer, Viewer, ReadOnly, and Unlicensed.

On a multi-site server, the command does not assign the user to a site. Therefore, the only site roles the command can successfully assign are ServerAdministrator and Unlicensed. If you specify any other site role, the command assigns the Unlicensed role.

On a single-site server, the user is created and added to the default site using the role that you specify.

If you have a user-based server installation, and if the command creates a new user but you have already reached the limit on the number of licenses for your users, the user is added as an unlicensed user.

--silent-progress

Do not display progress messages for the command.

Global options

The following options are used by all tabcmd commands. The --server, --user, and --password options are required at least once to begin a session. An authentication token is stored so subsequent commands can be run without including these options. This token remains valid for five minutes after the last command that used it.

-h, --help

Displays the help for the command.
Tableau Server on Linux Administrator Guide

-c, --use-certificate

Use client certificate to sign in. Required when mutual SSL is enabled.

For information about configuring the certificate, start with the following topic appropriate for your Tableau Server OS:

- **Windows:** Configure Mutual SSL
- **Linux:** Configure Mutual SSL

-s, --server

The Tableau Server URL, which is required at least once to begin session.

-u, --user

The Tableau Server username, which is required at least once to begin session.

-p, --password

The Tableau Server password, which is required at least once to begin session.

--password-file

Allows the password to be stored in the given .txt file rather than the command line for increased security.

-t, --site

Indicates that the command applies to the site specified by the Tableau Server site ID. If you do not specify a site, the Default site is assumed. Applies only to servers with multiple sites.

-x, --proxy

Host:Port
Uses the specified HTTP proxy.

--no-prompt

When specified, the command will not prompt for a password. If no valid password is provided the command will fail.

--no-proxy

When specified, an HTTP proxy will not be used.

--no-certcheck

When specified, tabcmd (the client) does not validate the server's SSL certificate.

--[no-]cookie

When specified, the session ID is saved on login so subsequent commands will not need to log in. Use the no- prefix to not save the session ID. By default the session is saved.

--timeout

Waits the specified number of seconds for the server to complete processing the command. By default the process will timeout in 30 seconds.

-

Specifies the end of options on the command line. You can use -- to indicate to tabcmd that anything that follows -- should not be interpreted as an option setting and can instead be interpreted as a value for the command. This is useful if you need to specify a value in the command that includes a hyphen. The following example shows how you might use -- in a tabcmd command, where -430105/Series1 is a required value for the export command.

```
tabcmd export --csv -f "D:\export10.csv" -- -430105/Series1
```
decryptextracts

Decrypt all extracts on a site. If no site is specified, extracts on the default site will be decrypted. For more information, see Extract Encryption at Rest.

Depending on the number and size of extracts, this operation may consume significant server resources. Consider running this command outside of normal business hours.

Example

tabcmd decryptextracts "West Coast Sales"

Global options

The following options are used by all tabcmd commands. The --server, --user, and --password options are required at least once to begin a session. An authentication token is stored so subsequent commands can be run without including these options. This token remains valid for five minutes after the last command that used it.

-h, --help

Displays the help for the command.

-c, --use-certificate

Use client certificate to sign in. Required when mutual SSL is enabled.

For information about configuring the certificate, start with the following topic appropriate for your Tableau Server OS:

- **Windows**: Configure Mutual SSL
- **Linux**: Configure Mutual SSL

-s, --server
The Tableau Server on Linux Administrator Guide

The Tableau Server URL, which is required at least once to begin session.

-u, --user

The Tableau Server username, which is required at least once to begin session.

-p, --password

The Tableau Server password, which is required at least once to begin session.

--password-file

Allows the password to be stored in the given .txt file rather than the command line for increased security.

-t, --site

Indicates that the command applies to the site specified by the Tableau Server site ID. If you do not specify a site, the Default site is assumed. Applies only to servers with multiple sites.

-x, --proxy

Host:Port

Uses the specified HTTP proxy.

--no-prompt

When specified, the command will not prompt for a password. If no valid password is provided the command will fail.

--no-proxy

When specified, an HTTP proxy will not be used.
--no-certcheck

When specified, tabcmd (the client) does not validate the server’s SSL certificate.

--[no-]cookie

When specified, the session ID is saved on login so subsequent commands will not need to log in. Use the no- prefix to not save the session ID. By default the session is saved.

--timeout

Waits the specified number of seconds for the server to complete processing the command. By default the process will timeout in 30 seconds.

--

Specifies the end of options on the command line. You can use -- to indicate to tabcmd that anything that follows -- should not be interpreted as an option setting and can instead be interpreted as a value for the command. This is useful if you need to specify a value in the command that includes a hyphen. The following example shows how you might use -- in a tabcmd command, where -430105/SHEET1 is a required value for the export command.

```
tabcmd export --csv -f "D:\export10.csv" -- -430105/SHEET1
```

**delete workbook-name or datasource-name**

Deletes the specified workbook or data source from the server.

This command takes the name of the workbook or data source as it is on the server, not the file name when it was published.

**Example**

```
tabcmd delete "Sales_Analysis"
```
Options

-r, --project

The name of the project containing the workbook or data source you want to delete. If not specified, the "Default" project is assumed.

--parent-project-path

Specifies the name of the parent project for the nested project as specified with the -r option. For example, to specify a project called "Nested" that exists in a "Main" project, use the following syntax: --parent-project-path "Main" -r "Nested".

--workbook

The name of the workbook you want to delete.

--datasource

The name of the data source you want to delete.

Global options

The following options are used by all tabcmd commands. The --server, --user, and --password options are required at least once to begin a session. An authentication token is stored so subsequent commands can be run without including these options. This token remains valid for five minutes after the last command that used it.

-h, --help

Displays the help for the command.

-c, --use-certificate
Use client certificate to sign in. Required when mutual SSL is enabled.

For information about configuring the certificate, start with the following topic appropriate for your Tableau Server OS:

- **Windows**: Configure Mutual SSL
- **Linux**: Configure Mutual SSL

- **-s, --server**
  
The Tableau Server URL, which is required at least once to begin session.

- **-u, --user**
  
The Tableau Server username, which is required at least once to begin session.

- **-p, --password**
  
The Tableau Server password, which is required at least once to begin session.

- **--password-file**
  
  Allows the password to be stored in the given .txt file rather than the command line for increased security.

- **-t, --site**
  
  Indicates that the command applies to the site specified by the Tableau Server site ID. If you do not specify a site, the Default site is assumed. Applies only to servers with multiple sites.

- **-x, --proxy**
  
  Host:Port
Uses the specified HTTP proxy.

--no-prompt

When specified, the command will not prompt for a password. If no valid password is provided the command will fail.

--no-proxy

When specified, an HTTP proxy will not be used.

--no-certcheck

When specified, tabcmd (the client) does not validate the server's SSL certificate.

--[no-]cookie

When specified, the session ID is saved on login so subsequent commands will not need to log in. Use the no- prefix to not save the session ID. By default the session is saved.

--timeout

Waits the specified number of seconds for the server to complete processing the command. By default the process will timeout in 30 seconds.

--

Specifies the end of options on the command line. You can use -- to indicate to tabcmd that anything that follows -- should not be interpreted as an option setting and can instead be interpreted as a value for the command. This is useful if you need to specify a value in the command that includes a hyphen. The following example shows how you might use -- in a tabcmd command, where -430105/SHEET1 is a required value for the export command.

```
tabcmd export --csv -f "D:\export10.csv" -- -430105/SHEET1
```
deleteextracts

Deletes extracts for a published workbook or data source.

Options

-d, --datasource

The name of the target data source for extract deletion.

--embedded-datasources

A space-separated list of embedded data source names within the target workbook. Enclose data source names with double quotes if they contain spaces. Only available when deleting extracts for a workbook.

--encrypt

Create encrypted extract.

--include-all

Include all embedded data sources within target workbook.

--parent-project-path

Path of the project that is the parent of the project that contains the target resource. Must specify the project name with --project.

--project

The name of the project that contains the target resource. Only necessary if --workbook or --datasource is specified. If unspecified, the default project 'Default' is used.

-u, -url

The canonical name for the resource as it appears in the URL.
-w, --workbook

The name of the target workbook for extract deletion.

Global options

The following options are used by all tabcmd commands. The --server, --user, and --password options are required at least once to begin a session. An authentication token is stored so subsequent commands can be run without including these options. This token remains valid for five minutes after the last command that used it.

-h, --help

Displays the help for the command.

-c, --use-certificate

Use client certificate to sign in. Required when mutual SSL is enabled.

For information about configuring the certificate, start with the following topic appropriate for your Tableau Server OS:

- **Windows**: [Configure Mutual SSL](#)
- **Linux**: [Configure Mutual SSL](#)

-s, --server

The Tableau Server URL, which is required at least once to begin session.

-u, --user

The Tableau Server username, which is required at least once to begin session.

-p, --password
The Tableau Server password, which is required at least once to begin session.

--password-file

Allows the password to be stored in the given .txt file rather than the command line for increased security.

-t, --site

Indicates that the command applies to the site specified by the Tableau Server site ID. If you do not specify a site, the Default site is assumed. Applies only to servers with multiple sites.

-x, --proxy

Host:Port

Uses the specified HTTP proxy.

--no-prompt

When specified, the command will not prompt for a password. If no valid password is provided the command will fail.

--no-proxy

When specified, an HTTP proxy will not be used.

--no-cert.check

When specified, tabcmd (the client) does not validate the server's SSL certificate.

--[no-]cookie

When specified, the session ID is saved on login so subsequent commands will not
need to log in. Use the no- prefix to not save the session ID. By default the session is saved.

--timeout

Waits the specified number of seconds for the server to complete processing the command. By default the process will timeout in 30 seconds.

--

Specifies the end of options on the command line. You can use -- to indicate to tabcmd that anything that follows -- should not be interpreted as an option setting and can instead be interpreted as a value for the command. This is useful if you need to specify a value in the command that includes a hyphen. The following example shows how you might use -- in a tabcmd command, where -430105/SHEET1 is a required value for the export command.

  tabcmd export --csv -f "D:\export10.csv" -- -430105/SHEET1

deletegroup group-name

Deletes the specified group from the server.

Example

tabcmd deletegroup "Development"

Global options

The following options are used by all tabcmd commands. The --server, --user, and --password options are required at least once to begin a session. An authentication token is stored so subsequent commands can be run without including these options. This token remains valid for five minutes after the last command that used it.
-h, --help

Displays the help for the command.

-c, --use-certificate

Use client certificate to sign in. Required when mutual SSL is enabled.

For information about configuring the certificate, start with the following topic appropriate for your Tableau Server OS:

- Windows: Configure Mutual SSL
- Linux: Configure Mutual SSL

-s, --server

The Tableau Server URL, which is required at least once to begin session.

-u, --user

The Tableau Server username, which is required at least once to begin session.

-p, --password

The Tableau Server password, which is required at least once to begin session.

--password-file

Allows the password to be stored in the given .txt file rather than the command line for increased security.

-t, --site

Indicates that the command applies to the site specified by the Tableau Server site ID. If you do not specify a site, the Default site is assumed. Applies only to servers with
multiple sites.

-\texttt{x, --proxy}

Host:Port

Uses the specified HTTP proxy.

-\texttt{--no-prompt}

When specified, the command will not prompt for a password. If no valid password is provided the command will fail.

-\texttt{--no-proxy}

When specified, an HTTP proxy will not be used.

-\texttt{--no-certcheck}

When specified, tabcmd (the client) does not validate the server's SSL certificate.

-\texttt{[no-]cookie}

When specified, the session ID is saved on login so subsequent commands will not need to log in. Use the \texttt{no-} prefix to not save the session ID. By default the session is saved.

-\texttt{--timeout}

Waits the specified number of seconds for the server to complete processing the command. By default the process will timeout in 30 seconds.

-\texttt{--}

Specifies the end of options on the command line. You can use -- to indicate to \texttt{tabcmd} that anything that follows -- should not be interpreted as an option setting and
can instead be interpreted as a value for the command. This is useful if you need to specify a value in the command that includes a hyphen. The following example shows how you might use -- in a `tabcmd` command, where `-430105/SHEET1` is a required value for the `export` command.

```shell
tabcmd export --csv -f "D:\export10.csv" -- -430105/SHEET1
```

`deleteproject` *project-name*

Deletes the specified project from the server.

Using `tabcmd`, you can specify only a top-level project in a project hierarchy. To automate tasks you want to perform on a project within a parent project, use the equivalent Tableau REST API call.

**Example**

```shell
tabcmd deleteproject "Designs"
```

**Option**

`--parent-project-path`

Specifies the name of the parent project for the nested project as specified with the command. For example, to specify a project called "Designs" that exists in a "Main" project, use the following syntax: `--parent-project-path "Main" "Designs"`.

**Global options**

The following options are used by all `tabcmd` commands. The `--server`, `--user`, and `--password` options are required at least once to begin a session. An authentication token is stored so subsequent commands can be run without including these options. This token remains valid for five minutes after the last command that used it.

```
-h, --help
```
Displays the help for the command.

-c, --use-certificate

Use client certificate to sign in. Required when mutual SSL is enabled.

For information about configuring the certificate, start with the following topic appropriate for your Tableau Server OS:

- **Windows**: Configure Mutual SSL
- **Linux**: Configure Mutual SSL

-s, --server

The Tableau Server URL, which is required at least once to begin session.

-u, --user

The Tableau Server username, which is required at least once to begin session.

-p, --password

The Tableau Server password, which is required at least once to begin session.

--password-file

Allows the password to be stored in the given .txt file rather than the command line for increased security.

-t, --site

Indicates that the command applies to the site specified by the Tableau Server site ID. If you do not specify a site, the Default site is assumed. Applies only to servers with multiple sites.
Tableau Server on Linux Administrator Guide

-x, --proxy

Host:Port

Uses the specified HTTP proxy.

--no-prompt

When specified, the command will not prompt for a password. If no valid password is provided the command will fail.

--no-proxy

When specified, an HTTP proxy will not be used.

--no-certcheck

When specified, tabcmd (the client) does not validate the server’s SSL certificate.

--[no-]cookie

When specified, the session ID is saved on login so subsequent commands will not need to log in. Use the no- prefix to not save the session ID. By default the session is saved.

--timeout

Waits the specified number of seconds for the server to complete processing the command. By default the process will timeout in 30 seconds.

--

Specifies the end of options on the command line. You can use -- to indicate to tabcmd that anything that follows -- should not be interpreted as an option setting and can instead be interpreted as a value for the command. This is useful if you need to specify a value in the command that includes a hyphen. The following example shows how you
might use -- in a `tabcmd` command, where `-430105/Sheet1` is a required value for the `export` command.

```
tabcmd export --csv -f "D:\export10.csv" -- -430105/Sheet1
```

deletesite `site-name`

Deletes the specified site from the server.

**Example**

```
tabcmd deletesite "Development"
```

**Global options**

The following options are used by all `tabcmd` commands. The `--server`, `--user`, and `--password` options are required at least once to begin a session. An authentication token is stored so subsequent commands can be run without including these options. This token remains valid for five minutes after the last command that used it.

- `h`, `--help`  
  Displays the help for the command.

- `c`, `--use-certificate`  
  Use client certificate to sign in. Required when mutual SSL is enabled.

For information about configuring the certificate, start with the following topic appropriate for your Tableau Server OS:

- **Windows**: Configure Mutual SSL
- **Linux**: Configure Mutual SSL
Tableau Server on Linux Administrator Guide

-s, --server

The Tableau Server URL, which is required at least once to begin session.

-u, --user

The Tableau Server username, which is required at least once to begin session.

-p, --password

The Tableau Server password, which is required at least once to begin session.

--password-file

Allows the password to be stored in the given .txt file rather than the command line for increased security.

-t, --site

Indicates that the command applies to the site specified by the Tableau Server site ID. If you do not specify a site, the Default site is assumed. Applies only to servers with multiple sites.

-x, --proxy

Host:Port

Uses the specified HTTP proxy.

--no-prompt

When specified, the command will not prompt for a password. If no valid password is provided the command will fail.

--no-proxy
When specified, an HTTP proxy will not be used.

--no-certcheck

When specified, tabcmd (the client) does not validate the server's SSL certificate.

--[no-]cookie

When specified, the session ID is saved on login so subsequent commands will not need to log in. Use the no- prefix to not save the session ID. By default the session is saved.

--timeout

Waits the specified number of seconds for the server to complete processing the command. By default the process will timeout in 30 seconds.

--

Specifies the end of options on the command line. You can use -- to indicate to tabcmd that anything that follows -- should not be interpreted as an option setting and can instead be interpreted as a value for the command. This is useful if you need to specify a value in the command that includes a hyphen. The following example shows how you might use -- in a tabcmd command, where -430105/Sheet1 is a required value for the export command.

```
tabcmd export --csv -f "D:\export10.csv" -- -430105/Sheet1
```

deletesiteusers filename.csv

Removes users from from the site that you are logged in to. The users to be removed are specified in a file that contains a simple list of one user name per line. (No additional information is required beyond the user name.)

By default, if the server has only one site, or if the user belongs to only one site, the user is also removed from the server. On a Tableau Server Enterprise installation, if the server
contains multiple sites, users who are assigned the site role of **Server Administrator** are removed from the site but are not removed from the server.

If the user owns content, the user's role is change to **Unlicensed**, but the user is not removed from the server or the site. The content is still owned by that user. To remove the user completely, you must change the owner of the content and then try removing the user again.

If the user was imported from Active Directory, the user is removed from the site and possibly from the server. However, the user is not deleted from Active Directory.

**Example**

```
tabcmd deletesiteusers "users.csv"
```

**Global options**

The following options are used by all `tabcmd` commands. The `--server`, `--user`, and `--password` options are required at least once to begin a session. An authentication token is stored so subsequent commands can be run without including these options. This token remains valid for five minutes after the last command that used it.

- **-h, --help**
  
  Displays the help for the command.

- **-c, --use-certificate**
  
  Use client certificate to sign in. Required when mutual SSL is enabled.

  For information about configuring the certificate, start with the following topic appropriate for your Tableau Server OS:
- **Windows**: Configure Mutual SSL
- **Linux**: Configure Mutual SSL

- **-s, --server**
  
The Tableau Server URL, which is required at least once to begin session.

- **-u, --user**
  
The Tableau Server username, which is required at least once to begin session.

- **-p, --password**
  
The Tableau Server password, which is required at least once to begin session.

- **--password-file**
  
  Allows the password to be stored in the given `.txt` file rather than the command line for increased security.

- **-t, --site**
  
  Indicates that the command applies to the site specified by the Tableau Server site ID. If you do not specify a site, the Default site is assumed. Applies only to servers with multiple sites.

- **-x, --proxy**
  
  Host:Port

  Uses the specified HTTP proxy.

- **--no-prompt**
  
  When specified, the command will not prompt for a password. If no valid password is
--no-proxy

When specified, an HTTP proxy will not be used.

--no-certcheck

When specified, tabcmd (the client) does not validate the server’s SSL certificate.

--[no-]cookie

When specified, the session ID is saved on login so subsequent commands will not need to log in. Use the no- prefix to not save the session ID. By default the session is saved.

--timeout

Waits the specified number of seconds for the server to complete processing the command. By default the process will timeout in 30 seconds.

--

Specifies the end of options on the command line. You can use -- to indicate to tabcmd that anything that follows -- should not be interpreted as an option setting and can instead be interpreted as a value for the command. This is useful if you need to specify a value in the command that includes a hyphen. The following example shows how you might use -- in a tabcmd command, where -430105/Sheet1 is a required value for the export command.

```bash
  tabcmd export --csv -f "D:\export10.csv" -- -430105/Sheet1
```

deleteusers *filename.csv*

Deletes the users listed in the specified comma-separated values (.csv) file.

The .csv file should contain a simple list of one user name per line.
Example

tabcmd deleteusers "users.csv"

Options

--[no-]complete

When set to --complete this option requires that all rows be valid for any change to succeed. If not specified, --complete is used.

Global options

The following options are used by all tabcmd commands. The --server, --user, and --password options are required at least once to begin a session. An authentication token is stored so subsequent commands can be run without including these options. This token remains valid for five minutes after the last command that used it.

-h, --help

Displays the help for the command.

-c, --use-certificate

Use client certificate to sign in. Required when mutual SSL is enabled.

For information about configuring the certificate, start with the following topic appropriate for your Tableau Server OS:

- Windows: Configure Mutual SSL

- Linux: Configure Mutual SSL

-s, --server
Tableau Server on Linux Administrator Guide

The Tableau Server URL, which is required at least once to begin session.

-u, --user

The Tableau Server username, which is required at least once to begin session.

-p, --password

The Tableau Server password, which is required at least once to begin session.

--password-file

Allows the password to be stored in the given .txt file rather than the command line for increased security.

-t, --site

Indicates that the command applies to the site specified by the Tableau Server site ID. If you do not specify a site, the Default site is assumed. Applies only to servers with multiple sites.

-x, --proxy

Host:Port

Uses the specified HTTP proxy.

--no-prompt

When specified, the command will not prompt for a password. If no valid password is provided the command will fail.

--no-proxy

When specified, an HTTP proxy will not be used.
--no-certcheck

When specified, tabcmd (the client) does not validate the server’s SSL certificate.

--[no-]cookie

When specified, the session ID is saved on login so subsequent commands will not need to log in. Use the no- prefix to not save the session ID. By default the session is saved.

--timeout

Waits the specified number of seconds for the server to complete processing the command. By default the process will timeout in 30 seconds.

--

Specifies the end of options on the command line. You can use -- to indicate to tabcmd that anything that follows -- should not be interpreted as an option setting and can instead be interpreted as a value for the command. This is useful if you need to specify a value in the command that includes a hyphen. The following example shows how you might use -- in a tabcmd command, where -430105/Sheet1 is a required value for the export command.

tabcmd export --csv -f "D:\export10.csv" -- -430105/Sheet1

editdomain

Changes the nickname or full domain name of an Active Directory domain on the server. A domain “nickname” is the Windows NetBIOS domain name.

You can modify the nickname for any domain the server is using. In general, you can modify the full domain name for any domain except the one that you used to sign in. However, if the user name that you are currently signed in with exists in both the current domain and the new domain, you can modify the full name for the current domain.
Review User Management in Active Directory Deployments to understand how multiple domains, domain name mapping, and user names interact with Tableau Server.

To see a list of domains, use listdomains.

**Examples**

```
tabcmd editdomain --id 2 --nickname "new-nickname"
```

```
tabcmd editdomain --id 3 --name "new-name"
```

**Options**

**--id**

The ID of domain to change. To get a list of domain IDs, use use listdomains.

**--name**

The new name for the domain.

**--nickname**

The new nickname for the domain.

**Global options**

The following options are used by all tabcmd commands. The `--server`, `--user`, and `--password` options are required at least once to begin a session. An authentication token is stored so subsequent commands can be run without including these options. This token remains valid for five minutes after the last command that used it.

**-h, --help**

Displays the help for the command.
Use client certificate to sign in. Required when mutual SSL is enabled.

For information about configuring the certificate, start with the following topic appropriate for your Tableau Server OS:

- **Windows**: Configure Mutual SSL
- **Linux**: Configure Mutual SSL

The Tableau Server URL, which is required at least once to begin session.

The Tableau Server username, which is required at least once to begin session.

The Tableau Server password, which is required at least once to begin session.

Allows the password to be stored in the given `.txt` file rather than the command line for increased security.

Indicates that the command applies to the site specified by the Tableau Server site ID. If you do not specify a site, the Default site is assumed. Applies only to servers with multiple sites.

Host:Port
Uses the specified HTTP proxy.

--no-prompt

When specified, the command will not prompt for a password. If no valid password is provided the command will fail.

--no-proxy

When specified, an HTTP proxy will not be used.

--no-certcheck

When specified, tabcmd (the client) does not validate the server's SSL certificate.

[--no-]cookie

When specified, the session ID is saved on login so subsequent commands will not need to log in. Use the no- prefix to not save the session ID. By default the session is saved.

--timeout

Waits the specified number of seconds for the server to complete processing the command. By default the process will timeout in 30 seconds.

--

Specifies the end of options on the command line. You can use -- to indicate to tabcmd that anything that follows -- should not be interpreted as an option setting and can instead be interpreted as a value for the command. This is useful if you need to specify a value in the command that includes a hyphen. The following example shows how you might use -- in a tabcmd command, where -430105/Sheet1 is a required value for the export command.

tabcmd export --csv -f "D:\export10.csv" -- -430105/Sheet1
editsite *site-name*

Changes the name of a site or its web folder name. You can also use this command to allow or deny site administrators the ability to add and remove users. If site administrators have user management rights, you can specify how many users they can add to a site.

**Examples**

```
tabcmd editsite wc_sales --site-name "West Coast Sales"
tabcmd editsite wc_sales --site-id "wsales"
tabcmd editsite wsales --status ACTIVE
tabcmd editsite wsales --user-quota 50
```

**Options**

--site-name

The name of the site that's displayed.

--site-id

Used in the URL to uniquely identify the site.

--user-quota

Maximum number of users who can be members of the site.

--[no-]site-mode

Allow or prevent site administrators from adding users to the site.

--status

Set to `ACTIVE` to activate a site, or to `SUSPENDED` to suspend a site.
In MB, the amount of workbooks, extracts, and data sources that can be stored on the site.

The extract encryption mode for the site can be enforced, enabled or disabled. For more information, see Extract Encryption at Rest. Depending on the number and size of extracts, this operation may consume significant server resources.

Global options

The following options are used by all tabcmd commands. The --server, --user, and --password options are required at least once to begin a session. An authentication token is stored so subsequent commands can be run without including these options. This token remains valid for five minutes after the last command that used it.

-h, --help

Displays the help for the command.

-c, --use-certificate

Use client certificate to sign in. Required when mutual SSL is enabled.

For information about configuring the certificate, start with the following topic appropriate for your Tableau Server OS:

- Windows: Configure Mutual SSL
- Linux: Configure Mutual SSL

-s, --server
The Tableau Server URL, which is required at least once to begin session.

-u, --user

The Tableau Server username, which is required at least once to begin session.

-p, --password

The Tableau Server password, which is required at least once to begin session.

--password-file

Allows the password to be stored in the given .txt file rather than the command line for increased security.

-t, --site

Indicates that the command applies to the site specified by the Tableau Server site ID. If you do not specify a site, the Default site is assumed. Applies only to servers with multiple sites.

-x, --proxy

Host:Port

Uses the specified HTTP proxy.

--no-prompt

When specified, the command will not prompt for a password. If no valid password is provided the command will fail.

--no-proxy

When specified, an HTTP proxy will not be used.
--no-certcheck

When specified, tabcmd (the client) does not validate the server’s SSL certificate.

--[no-]cookie

When specified, the session ID is saved on login so subsequent commands will not need to log in. Use the no- prefix to not save the session ID. By default the session is saved.

--timeout

Waits the specified number of seconds for the server to complete processing the command. By default the process will timeout in 30 seconds.

--

Specifies the end of options on the command line. You can use -- to indicate to tabcmd that anything that follows -- should not be interpreted as an option setting and can instead be interpreted as a value for the command. This is useful if you need to specify a value in the command that includes a hyphen. The following example shows how you might use -- in a tabcmd command, where -430105/Ser... is a required value for the export command.

tabcmd export --csv -f "D:\export10.csv" -- -430105/Ser...

encryptextracts

Encrypt all extracts on a site. If no site is specified, extracts on the default site will be encrypted. For more information, see Extract Encryption at Rest.

Depending on the number and size of extracts, this operation may consume significant server resources. Consider running this command outside of normal business hours.

Example

tabcmd encryptextracts "West Coast Sales"
Global options

The following options are used by all tabcmd commands. The `--server`, `--user`, and `--password` options are required at least once to begin a session. An authentication token is stored so subsequent commands can be run without including these options. This token remains valid for five minutes after the last command that used it.

`-h, --help`

Displays the help for the command.

`-c, --use-certificate`

Use client certificate to sign in. Required when mutual SSL is enabled.

For information about configuring the certificate, start with the following topic appropriate for your Tableau Server OS:

- **Windows**: Configure Mutual SSL
- **Linux**: Configure Mutual SSL

`-s, --server`

The Tableau Server URL, which is required at least once to begin session.

`-u, --user`

The Tableau Server username, which is required at least once to begin session.

`-p, --password`

The Tableau Server password, which is required at least once to begin session.

`--password-file`
Tableau Server on Linux Administrator Guide

Allows the password to be stored in the given .txt file rather than the command line for increased security.

-t, --site

Indicates that the command applies to the site specified by the Tableau Server site ID. If you do not specify a site, the Default site is assumed. Applies only to servers with multiple sites.

-x, --proxy

Host:Port

Uses the specified HTTP proxy.

--no-prompt

When specified, the command will not prompt for a password. If no valid password is provided the command will fail.

--no-proxy

When specified, an HTTP proxy will not be used.

--no-certcheck

When specified, tabcmd (the client) does not validate the server's SSL certificate.

--[no-]cookie

When specified, the session ID is saved on login so subsequent commands will not need to log in. Use the no- prefix to not save the session ID. By default the session is saved.

--timeout
Waits the specified number of seconds for the server to complete processing the command. By default the process will timeout in 30 seconds.

`--`

Specifies the end of options on the command line. You can use `--` to indicate to `tabcmd` that anything that follows `--` should not be interpreted as an option setting and can instead be interpreted as a value for the command. This is useful if you need to specify a value in the command that includes a hyphen. The following example shows how you might use `--` in a `tabcmd command, where `-430105/Sheet1` is a required value for the `export` command.

```
export --csv -f "D:\export10.csv" -- -430105/Sheet1
```

**export**

Exports a view or workbook from Tableau Server and saves it to a file. This command can also export just the data used for a view.

Note the following when you use this command:

- **Permissions**: To export, you must have the **Export Image** permission. By default, this permission is Allowed or Inherited for all roles, although permissions can be set per workbook or view.

- **Exporting data**: To export just the data for a view, use the `--csv` option. This exports the summary data used in a view to a `.csv` file.

- **Specifying the view, workbook, or data to export**:

  ```
  Use part of the URL to identify what to export, specifically the "work-book/view" string as it appears in the URL for the workbook or view. Do not use the "friendly name," and exclude the :iid=<n> session ID at the end of the URL.
  ```
For example, the Tableau sample view *Global Temperatures* in the *Regional* workbook has a URL similar to this: `<server_name>/#/views/Regional/GlobalTemperatures?:iid=3`

To export the *Global Temperatures* view, use the string *Regional/GlobalTemperatures*.

Do *not* use *Regional/Global Temperatures*, or *Regional/GlobalTemperatures?:iid=3*.

- If the server is running multiple sites and the view or workbook is on a site other than Default, use `-t <site_id>`.

- To export a workbook, get the URL string by opening a view in the workbook, and include the view in the string you use.

  In the above example, to export the *Regional* workbook, use the string *Regional/GlobalTemperatures*.

- To export a workbook, it must have been published with **Show Sheets as Tabs** selected in the Tableau Desktop Publish dialog box.

  **Note:** The Tableau workbook that contains the administrative views cannot be exported.

- **The saved file's format:** Your format options depend on what's being exported. A workbook can only be exported as a PDF using the `--fullpdf` argument. A view can be exported as a PDF (`--pdf`) or a PNG (`--png`).

- **The saved file's name and location** (optional): If you don't provide a name, it will be derived from the view or workbook name. If you don't provide a location, the file will be saved to your current working directory. Otherwise, you can specify a full path or one that's relative to your current working directory.
Note: You must include a file name extension such as .csv or .pdf. The command does not automatically add an extension to the file name that you provide.

- **Dashboard web page objects not included in PDF exports:** A dashboard can optionally include a web page object. If you are performing an export to PDF of a dashboard that includes a web page object, the web page object won't be included in the PDF.

- **Non-ASCII and non-standard ASCII characters and PDF exports:** If you are exporting a view or workbook with a name that includes a character outside the ASCII character set, or a non-standard ASCII character set, you need to URL encode (percent-encode) the character.

  For example if your command includes the city Zürich, you need to URL encode it as Z%C3%BCrich:

  ```
  tabcmd export "Cities/Sheet1?locationCity=Z%C3%BCrich" -fullpdf
  ```

**Clearing the Cache to Use Real-Time Data**

You can optionally add the URL parameter `?:refresh=yes` to force a fresh data query instead of pulling the results from the cache. If you are using `tabcmd` with your own scripting and the `refresh` URL parameter is being used a great deal, this can have a negative impact on performance. It's recommended that you use `refresh` only when real-time data is required—for example, on a single dashboard instead of on an entire workbook.

**Examples**

**Views**

```bash
  tabcmd export "Q1Sales/Sales_Report" --csv -f "Weekly-Report.csv"
```
Tableau Server on Linux Administrator Guide

```bash
  tabcmd export -t Sales "Sales/Sales_Analysis" --pdf -f "C:\Tableau_Workbooks\Weekly-Reports.pdf"

  tabcmd export "Finance/InvestmentGrowth" --png

  tabcmd export "Finance/InvestmentGrowth?:refresh=yes" --png

  Workbooks

  tabcmd export "Q1Sales/Sales_Report" --fullpdf

  tabcmd export "Sales/Sales_Analysis" --fullpdf --pagesize tabloid -f "C:\Tableau_Workbooks\Weekly-Reports.pdf"

  Options

  -f, --filename

    Saves the file with the given filename and extension.

  --csv

    View only. Export the view's data (summary data) in .csv format.

  --pdf

    View only. Export as a PDF.

  --png

    View only. Export as an image in .png format.

  --fullpdf

    Workbook only. Export as a PDF. The workbook must have been published with Show Sheets as Tabs enabled.
```
--pagelayout

Sets the page orientation (landscape or portrait) of the exported PDF. If not specified, its Tableau Desktop setting will be used.

--pagesize

Sets the page size of the exported PDF as one of the following: unspecified, letter, legal, note, folio, tabloid, ledger, statement, executive, a3, a4, a5, b4, b5, or quarto. Default is letter.

--width

Sets the width in pixels. Default is 800 px.

--height

Sets the height in pixels. Default is 600 px.

Global options

The following options are used by all tabcmd commands. The --server, --user, and --password options are required at least once to begin a session. An authentication token is stored so subsequent commands can be run without including these options. This token remains valid for five minutes after the last command that used it.

-h, --help

Displays the help for the command.

-c, --use-certificate

Use client certificate to sign in. Required when mutual SSL is enabled.
For information about configuring the certificate, start with the following topic appropriate for your Tableau Server OS:

- **Windows:** Configure Mutual SSL
- **Linux:** Configure Mutual SSL

```
-s, --server
```

The Tableau Server URL, which is required at least once to begin session.

```
-u, --user
```

The Tableau Server username, which is required at least once to begin session.

```
-p, --password
```

The Tableau Server password, which is required at least once to begin session.

```
--password-file
```

Allows the password to be stored in the given .txt file rather than the command line for increased security.

```
-t, --site
```

Indicates that the command applies to the site specified by the Tableau Server site ID. If you do not specify a site, the Default site is assumed. Applies only to servers with multiple sites.

```
-x, --proxy
```

Host:Port

Uses the specified HTTP proxy.
--no-prompt

When specified, the command will not prompt for a password. If no valid password is provided the command will fail.

--no-proxy

When specified, an HTTP proxy will not be used.

--no-certcheck

When specified, tabcmd (the client) does not validate the server’s SSL certificate.

--[no-]cookie

When specified, the session ID is saved on login so subsequent commands will not need to log in. Use the no- prefix to not save the session ID. By default the session is saved.

--timeout

Waits the specified number of seconds for the server to complete processing the command. By default the process will timeout in 30 seconds.

--

Specifies the end of options on the command line. You can use -- to indicate to tabcmd that anything that follows -- should not be interpreted as an option setting and can instead be interpreted as a value for the command. This is useful if you need to specify a value in the command that includes a hyphen. The following example shows how you might use -- in a tabcmd command, where -430105/Sheet1 is a required value for the export command.

```
tabcmd export --csv -f "D:\export10.csv" -- -430105/Sheet1
```
get url

Gets the resource from Tableau Server that's represented by the specified (partial) URL. The result is returned as a file.

Note the following when you use this command:

- **Permissions:** To get a file, you must have the **Download/Web Save As** permission. By default, this permission is allowed or inherited for all roles, although permissions can be set per workbook or view.

- **Specifying a view or workbook to get:** You specify a view to get using the "/views/<workbookname>/<viewname>.<extension>" string, and specify a workbook to get using the "/workbooks/<workbookname>.<extension>" string. Replace <workbookname> and <viewname> with the names of the workbook and view as they appear in the URL when you open the view in a browser and replace <extension> with the type of file you want to save. Do not use the session ID at the end of the URL (?:iid=<n>) or the "friendly" name of the workbook or view.

  For example, when you open a view **Regional Totals** in a workbook named **Metrics Summary**, the URL will look similar to this:

  /views/MetricsSummary_1/RegionalTotals?:iid=1

  Use the string /views/MetricsSummary_1/RegionalTotals.<extension> to get the view.

  Use the string /workbooks/MetricsSummary_1.<extension> to get the workbook.

- **File extension:** The URL must include a file extension. The extension determines what's returned. A view can be returned in PDF, PNG, or CSV (summary data only) format. A Tableau workbook is returned as a TWB if it connects to a published data source or uses a live connection, or a TWBX if it connects to a data extract.
Note: If you are downloading a view to a PDF or PNG file, and if you include a \( --filename \) parameter that includes the .pdf or .png extension, you do not have to include a .pdf or .png extension in the URL.

- **The saved file's name and location** (optional): The name you use for \( --filename \) should include the file extension. If you don't provide a name and file extension, both will be derived from the URL string. If you don't provide a location, the file is saved to your current working directory. Otherwise, you can specify a full path or one that's relative to your current working directory.

- **PNG size** (optional): If the saved file is a PNG, you can specify the size, in pixels, in the URL.

**Clearing the cache to use real-time data**

You can optionally add the URL parameter \( ?:refresh=yes \) to force a fresh data query instead of pulling the results from the cache. If you are using tabcmd with your own scripting, using the \( refresh \) parameter a great deal can have a negative impact on performance. It's recommended that you use \( refresh \) only when real-time data is required—for example, on a single dashboard instead of on an entire workbook.

**Examples**

**Views**

```bash
tabcmd get "/views/Sales_Analysis/Sales_Report.png" --filename "Weekly-Report.png"

tabcmd get "/views/Finance/InvestmentGrowth.pdf" -f "Q1Growth.pdf"

tabcmd get "/views/Finance/InvestmentGrowth" -f "Q1Growth.pdf"

tabcmd get "/views/Finance/InvestmentGrowth.csv"
```
Global options

The following options are used by all `tabcmd` commands. The `--server`, `--user`, and `--password` options are required at least once to begin a session. An authentication token is stored so subsequent commands can be run without including these options. This token remains valid for five minutes after the last command that used it.

- `--help`

  Displays the help for the command.

- `--use-certificate`

  Use client certificate to sign in. Required when mutual SSL is enabled.

  For information about configuring the certificate, start with the following topic appropriate for your Tableau Server OS:

  - **Windows**: Configure Mutual SSL
  - **Linux**: Configure Mutual SSL

- `--server`

  The Tableau Server URL, which is required at least once to begin session.
-u, --user

The Tableau Server username, which is required at least once to begin session.

-p, --password

The Tableau Server password, which is required at least once to begin session.

--password-file

Allows the password to be stored in the given .txt file rather than the command line for increased security.

-t, --site

Indicates that the command applies to the site specified by the Tableau Server site ID. If you do not specify a site, the Default site is assumed. Applies only to servers with multiple sites.

-x, --proxy

Host:Port

Uses the specified HTTP proxy.

--no-prompt

When specified, the command will not prompt for a password. If no valid password is provided, the command will fail.

--no-proxy

When specified, an HTTP proxy will not be used.

--no-certcheck
When specified, `tabcmd` (the client) does not validate the server's SSL certificate.

`--[no-]cookie`

When specified, the session ID is saved on login so subsequent commands will not need to log in. Use the `no-` prefix to not save the session ID. By default the session is saved.

`--timeout`

Waits the specified number of seconds for the server to complete processing the command. By default the process will timeout in 30 seconds.

`--`

Specifies the end of options on the command line. You can use `--` to indicate to `tabcmd` that anything that follows `--` should not be interpreted as an option setting and can instead be interpreted as a value for the command. This is useful if you need to specify a value in the command that includes a hyphen. The following example shows how you might use `--` in a `tabcmd` command, where `-430105/Sheet1` is a required value for the `export` command.

```
tabcmd export --csv -f "D:\export10.csv" -- -430105/Sheet1
```

`initialuser`

Create the initial administrative user on a server that does not have an initial administrative user defined.

Enclose values in single quotes.

**Note:** The `tabcmd initialuser` command does not require authentication to Tableau Server, but you must run the command on the initial server node.

**Examples**
tabcmd initialuser --username 'admin' --password 'password' --server http://localhost

To prompt for the password in the shell, do not include the --password parameter in the command. For example:

`tabcmd initialuser --username 'admin' --server http://localhost`

Options

-f, --friendly

Creates the initial administrative user with the display name.

Global options

The following options are used by all `tabcmd` commands. The --server, --user, and --password options are required at least once to begin a session. An authentication token is stored so subsequent commands can be run without including these options. This token remains valid for five minutes after the last command that used it.

-h, --help

Displays the help for the command.

-c, --use-certificate

Use client certificate to sign in. Required when mutual SSL is enabled.
For information about configuring the certificate, start with the following topic appropriate for your Tableau Server OS:

- **Windows**: Configure Mutual SSL
- **Linux**: Configure Mutual SSL

-s, --server

The Tableau Server URL, which is required at least once to begin session.

-u, --user

The Tableau Server username, which is required at least once to begin session.

-p, --password

The Tableau Server password, which is required at least once to begin session.

--password-file

Allows the password to be stored in the given .txt file rather than the command line for increased security.

-t, --site

Indicates that the command applies to the site specified by the Tableau Server site ID. If you do not specify a site, the Default site is assumed. Applies only to servers with multiple sites.

-x, --proxy

Host:Port

Uses the specified HTTP proxy.
--no-prompt

When specified, the command will not prompt for a password. If no valid password is provided the command will fail.

--no-proxy

When specified, an HTTP proxy will not be used.

--no-certcheck

When specified, tabcmd (the client) does not validate the server's SSL certificate.

--[no-]cookie

When specified, the session ID is saved on login so subsequent commands will not need to log in. Use the no- prefix to not save the session ID. By default the session is saved.

--timeout

Waits the specified number of seconds for the server to complete processing the command. By default the process will timeout in 30 seconds.

--

Specifies the end of options on the command line. You can use -- to indicate to tabcmd that anything that follows -- should not be interpreted as an option setting and can instead be interpreted as a value for the command. This is useful if you need to specify a value in the command that includes a hyphen. The following example shows how you might use -- in a tabcmd command, where -430105/SHEET1 is a required value for the export command.

tabcmd export --csv -f "D:\export10.csv" -- -430105/SHEET1
listdomains

Displays a list of the Active Directory domains that are in use on the server, along with their nicknames and IDs. If the server is configured to use local authentication, the command returns only the domain name local.

Example

tabcmd listdomains

Global options

The following options are used by all tabcmd commands. The --server, --user, and --password options are required at least once to begin a session. An authentication token is stored so subsequent commands can be run without including these options. This token remains valid for five minutes after the last command that used it.

-h, --help

Displays the help for the command.

-c, --use-certificate

Use client certificate to sign in. Required when mutual SSL is enabled.

For information about configuring the certificate, start with the following topic appropriate for your Tableau Server OS:

- **Windows**: Configure Mutual SSL
- **Linux**: Configure Mutual SSL

-s, --server

The Tableau Server URL, which is required at least once to begin session.
-u, --user

The Tableau Server username, which is required at least once to begin session.

-p, --password

The Tableau Server password, which is required at least once to begin session.

--password-file

Allows the password to be stored in the given .txt file rather than the command line for increased security.

-t, --site

Indicates that the command applies to the site specified by the Tableau Server site ID. If you do not specify a site, the Default site is assumed. Applies only to servers with multiple sites.

-x, --proxy

Host:Port

Uses the specified HTTP proxy.

--no-prompt

When specified, the command will not prompt for a password. If no valid password is provided the command will fail.

--no-proxy

When specified, an HTTP proxy will not be used.

--no-certcheck
When specified, tabcmd (the client) does not validate the server’s SSL certificate.

```
--[no-]cookie
```

When specified, the session ID is saved on login so subsequent commands will not need to log in. Use the `no-` prefix to not save the session ID. By default the session is saved.

```
--timeout
```

Waits the specified number of seconds for the server to complete processing the command. By default the process will timeout in 30 seconds.

```
-
```

Specifies the end of options on the command line. You can use `--` to indicate to `tabcmd` that anything that follows `--` should not be interpreted as an option setting and can instead be interpreted as a value for the command. This is useful if you need to specify a value in the command that includes a hyphen. The following example shows how you might use `--` in a `tabcmd` command, where `-430105/SHEET1` is a required value for the `export` command.

```
tabcmd export --csv -f "D:\export10.csv" -- -430105/SHEET1
```

listsites

Returns a list of sites to which the logged in user belongs.

Example

```
tabcmd listsites --username adam --password mypassword
```

Options

```
--get-extract-encryption-mode
```

The extract encryption mode for the site can be `enforced`, `enabled` or `disabled`. For
more information, see Extract Encryption at Rest.

Global options

The following options are used by all `tabcmd` commands. The `--server`, `--user`, and `--password` options are required at least once to begin a session. An authentication token is stored so subsequent commands can be run without including these options. This token remains valid for five minutes after the last command that used it.

- `--help`
  Displays the help for the command.

- `--use-certificate`
  Use client certificate to sign in. Required when mutual SSL is enabled.
  
  For information about configuring the certificate, start with the following topic appropriate for your Tableau Server OS:
  
  - **Windows**: Configure Mutual SSL
    
  - **Linux**: Configure Mutual SSL

- `--server`
  The Tableau Server URL, which is required at least once to begin session.

- `--user`
  The Tableau Server username, which is required at least once to begin session.

- `--password`
  The Tableau Server password, which is required at least once to begin session.
---password-file

Allows the password to be stored in the given .txt file rather than the command line for increased security.

-t, --site

Indicates that the command applies to the site specified by the Tableau Server site ID. If you do not specify a site, the Default site is assumed. Applies only to servers with multiple sites.

-x, --proxy

Host:Port

Uses the specified HTTP proxy.

--no-prompt

When specified, the command will not prompt for a password. If no valid password is provided the command will fail.

--no-proxy

When specified, an HTTP proxy will not be used.

--no-certcheck

When specified, tabcmd (the client) does not validate the server's SSL certificate.

--[no-]cookie

When specified, the session ID is saved on login so subsequent commands will not need to log in. Use the no- prefix to not save the session ID. By default the session is saved.
--timeout

Waits the specified number of seconds for the server to complete processing the command. By default the process will timeout in 30 seconds.

--

Specifies the end of options on the command line. You can use -- to indicate to tabcmd that anything that follows -- should not be interpreted as an option setting and can instead be interpreted as a value for the command. This is useful if you need to specify a value in the command that includes a hyphen. The following example shows how you might use -- in a tabcmd command, where -430105/Sheet1 is a required value for the export command.

tabcmd export --csv -f "D:\export10.csv" -- -430105/Sheet1

login

Logs in a Tableau Server user.

Use the --server,--site,--username,--password global options to create a session.

Note: When you use the tabcmd login command, you cannot use SAML single sign-on (SSO), even if the server is configured to use SAML. To log in, you must pass the user name and password of a user who has been created on the server. You will have the permissions of the Tableau Server user that you’re signed in as. For more information, see Set Users’ Site Roles and Permissions.

If you want to log in using the same information you’ve already used to create a session, just specify the --password option. The server and user name stored in the cookie will be used.

If the server is using a port other than 80 (the default), you will need to specify the port.
You need the --site (-t) option only if the server is running multiple sites and you are logging in to a site other than the Default site. If you do not provide a password you will be prompted for one. If the --no-prompt option is specified and no password is provided the command will fail.

Once you log in, the session will continue until it expires on the server or the logout command is run.

Example

Logs user jsmith in to the Tableau Server running on their local machine:

```
tabcmd login -s http://localhost -u jsmith -p password
```

Logs administrator in to the Sales site on sales-server:

```
tabcmd login -s http://sales-server -t Sales -u administrator -p password

   tabcmd login -s http://sales-server:8000 -t Sales -u administrator -p password
```

Logs administrator in to the Sales site on sales-server using SSL, but does not validate the server’s SSL certificate:

```
tabcmd login --no-certificate -s https://sales-server -t Sales -u administrator -p password
```

Establishes a forward proxy and port for localhost:

```
tabcmd login --proxy myfwdproxyserver:8888 -s http://localhost -u jsmith -p password
```

Logs user jsmith in to the reverse proxy using SSL:

```
tabcmd login -s https://myreverseproxy -u jsmith -p password
```
Options

-s, --server

If you are running the command from a Tableau Server computer that's on your network, you can use http://localhost. Otherwise, specify the computer's URL, such as http://bigbox.myco.com or http://bigbox.

For Tableau Online specify the URL https://online.tableau.com.

-t, --site

Include this option if the server has multiple sites, and you are logging in to a site other than the default site.

The site ID is used in the URL to uniquely identify the site. For example, a site named West Coast Sales might have a site ID of west-coast-sales.

-u, --username

The user name of the user logging in. For Tableau Online, the user name is the user's email address.

-p, --password

Password for the user specified for --username. If you do not provide a password you will be prompted for one.

--password-file

Allows the password to be stored in the given filename.txt file rather than the command line, for increased security.

-x, --proxy

Use to specify the HTTP proxy server and port (Host:Port) for the tabcmd request.
No prompt

Do not prompt for a password. If no password is specified, the login command will fail.

-no-proxy

Do not use an HTTP proxy server.

-cookie

Saves the session ID on login. Subsequent commands will not require a login. This value is the default for the command.

-no-cookie

Do not save the session ID information after a successful login. Subsequent commands will require a login.

-timeout SECONDS

The number of seconds the server should wait before processing the login command. Default: 30 seconds.

Global options

The following options are used by all tabcmd commands. The --server, --user, and --password options are required at least once to begin a session. An authentication token is stored so subsequent commands can be run without including these options. This token remains valid for five minutes after the last command that used it.

-h, --help

Displays the help for the command.
-c, --use-certificate

Use client certificate to sign in. Required when mutual SSL is enabled.

For information about configuring the certificate, start with the following topic appropriate for your Tableau Server OS:

- **Windows**: Configure Mutual SSL
- **Linux**: Configure Mutual SSL

-s, --server

The Tableau Server URL, which is required at least once to begin session.

-u, --user

The Tableau Server username, which is required at least once to begin session.

-p, --password

The Tableau Server password, which is required at least once to begin session.

--password-file

Allows the password to be stored in the given .txt file rather than the command line for increased security.

-t, --site

Indicates that the command applies to the site specified by the Tableau Server site ID. If you do not specify a site, the Default site is assumed. Applies only to servers with multiple sites.

-x, --proxy

Host:Port
Tableau Server on Linux Administrator Guide

Uses the specified HTTP proxy.

--no-prompt

When specified, the command will not prompt for a password. If no valid password is provided the command will fail.

--no-proxy

When specified, an HTTP proxy will not be used.

--no-certcheck

When specified, tabcmd (the client) does not validate the server’s SSL certificate.

--[no-]cookie

When specified, the session ID is saved on login so subsequent commands will not need to log in. Use the no- prefix to not save the session ID. By default the session is saved.

--timeout

Waits the specified number of seconds for the server to complete processing the command. By default the process will timeout in 30 seconds.

--

Specifies the end of options on the command line. You can use -- to indicate to tabcmd that anything that follows -- should not be interpreted as an option setting and can instead be interpreted as a value for the command. This is useful if you need to specify a value in the command that includes a hyphen. The following example shows how you might use -- in a tabcmd command, where -430105/Sheet1 is a required value for the export command.

tabcmd export --csv -f "D:\export10.csv" -- -430105/Sheet1
logout

Logs out of the server.

Example
tabcmd logout

publish *filename.twb(x), filename.tds(x), or filename.hyper*

Publishes the specified workbook (.twb(x)), data source (.tds(x)), or extract (.hyper) to Tableau Server.

If you are publishing a workbook, by default, all sheets in the workbook are published without database user names or passwords.

The permissions initially assigned to the workbook or data source are copied from the project that the file is published to. Permissions for the published resource can be changed after the file has been published.

If the workbook contains user filters, one of the thumbnail options must be specified.

Example
tabcmd publish "analysis.twbx" -n "Sales_Analysis" --db-username "jsmith" --db-password "secret-password"

tabcmd publish "analysis_sfdc.hyper" -n "Sales Analysis" --oauth-username "user-name" --save-oauth

If the file is not in the same directory as tabcmd, include the full path to the file.

Example
tabcmd publish "\\computer\volume\Tableau Workbooks\analysis.twbx" -n "Sales_Analysis" --db-username "jsmith" --db-password "secret-password"
Tableau Server on Linux Administrator Guide

tabcmd publish "\\computer\volume\Tableau Workbooks\analysis_sfd-c.hyper" -n "Sales Analysis" --oauth-username "username" --save-oauth

Options

-n, --name

Name of the workbook or data source on the server. If omitted, the workbook, data source, or data extract will be named after filename.

-o, --overwrite

Overwrites the workbook, data source, or data extract if it already exists on the server.

-r, --project

Publishes the workbook, data source, or data extract into the specified project. Publishes to the "Default" project if not specified.

--parent-project-path

Specifies the name of the parent project for the nested project as specified with the -r option. For example, to specify a project called "Nested" that exists in a "Main" project, use the following syntax: --parent-project-path "Main" -r "Nested".

--db-username

Use this option to publish a database user name with the workbook, data source, or data extract.

--db-password

Use this option to publish a database password with the workbook, data source, or data extract.
--save-db-password

Stores the provided database password on the server.

--oauth-username

The email address of the user account. Connects the user through a preconfigured OAuth connection, if the user already has a saved access token for the cloud data source specified in --name. Access tokens are managed in user preferences.

For existing OAuth connections to the data source, use this option instead of --db-username and --db-password.

--save-oauth

Saves the credential specified by --oauth-username as an embedded credential with the published workbook or data source.

Subsequently, when the publisher or server administrator signs in to the server and edits the connection for that workbook or data source, the connection settings will show this OAuth credential as embedded in the content.

If you want to schedule extract refreshes after publishing, you must include this option with --oauth-username. This is analogous to using --save-db-password with a traditional database connection.

--thumbnail-username

If the workbook contains user filters, the thumbnails will be generated based on what the specified user can see. Cannot be specified when --thumbnail-group option is set.

--thumbnail-group

If the workbook contains user filters, the thumbnails will be generated based on what
the specified group can see. Cannot be specified when \texttt{--thumbnail-username} option is set.

\texttt{--tabbed}

When a workbook with tabbed views is published, each sheet becomes a tab that viewers can use to navigate through the workbook. Note that this setting will override any sheet-level security.

\texttt{--append}

Append the extract file to the existing data source.

\texttt{--replace}

Use the extract file to replace the existing data source.

\texttt{--disable-uploader}

Disable the incremental file uploader.

\texttt{--restart}

Restart the file upload.

\texttt{--encrypt-extracts}

Encrypt extracts when you publish a workbook, data source, or extract to the server. For more information, see Extract Encryption at Rest.

Global options

The following options are used by all \texttt{tabcmd} commands. The \texttt{--server}, \texttt{--user}, and \texttt{--password} options are required at least once to begin a session. An authentication token is
stored so subsequent commands can be run without including these options. This token remains valid for five minutes after the last command that used it.

-\( h, \--\text{help} \)

Displays the help for the command.

-\( c, \--\text{use-certificate} \)

Use client certificate to sign in. Required when mutual SSL is enabled.

For information about configuring the certificate, start with the following topic appropriate for your Tableau Server OS:

- Windows: Configure Mutual SSL
- Linux: Configure Mutual SSL

-\( s, \--\text{server} \)

The Tableau Server URL, which is required at least once to begin session.

-\( u, \--\text{user} \)

The Tableau Server username, which is required at least once to begin session.

-\( p, \--\text{password} \)

The Tableau Server password, which is required at least once to begin session.

-\( \text{password-file} \)

Allows the password to be stored in the given .txt file rather than the command line for increased security.

-\( t, \--\text{site} \)
Indicates that the command applies to the site specified by the Tableau Server site ID. If you do not specify a site, the Default site is assumed. Applies only to servers with multiple sites.

-x, --proxy

Host:Port

Uses the specified HTTP proxy.

--no-prompt

When specified, the command will not prompt for a password. If no valid password is provided the command will fail.

--no-proxy

When specified, an HTTP proxy will not be used.

--no-certcheck

When specified, tabcmd (the client) does not validate the server’s SSL certificate.

--[no-]cookie

When specified, the session ID is saved on login so subsequent commands will not need to log in. Use the no- prefix to not save the session ID. By default the session is saved.

--timeout

Waits the specified number of seconds for the server to complete processing the command. By default the process will timeout in 30 seconds.
Specifies the end of options on the command line. You can use -- to indicate to \texttt{tabcmd} that anything that follows -- should not be interpreted as an option setting and can instead be interpreted as a value for the command. This is useful if you need to specify a value in the command that includes a hyphen. The following example shows how you might use -- in a \texttt{tabcmd} command, where \texttt{-430105/Sheet1} is a required value for the \texttt{export} command.

\texttt{tabcmd export --csv -f "D:\export10.csv" -- -430105/Sheet1}

\texttt{publishsamples}

\textbf{Description}

Publishes Tableau Sample workbooks to the specified project. Any existing samples will be overwritten.

\textbf{Syntax}

\texttt{tabcmd publishsamples -n [project name] [Global options]}

\textbf{Example}

Publish samples to the Inside Sales project on the Default site, as user jsmith.

\texttt{tabcmd publishsamples -n "Inside Sales" -t "" -s localhost -- username "jsmith" --password "secret-password"}

\textbf{Options}

\texttt{-n, --name}

Required. Publishes the Tableau samples into the specified project. If the project name includes spaces, enclose the entire name in quotes.

\texttt{--parent-project-path}

Specifies the name of the parent project for the nested project as specified with the \texttt{-n} option. For example, to specify a project called "Nested" that exists in a "Main"
Tableau Server on Linux Administrator Guide

To create a nested project, use the following syntax: `--parent-project-path "Main" -n "Nested"`.

Global options

The following options are used by all `tabcmd` commands. The `--server`, `--user`, and `--password` options are required at least once to begin a session. An authentication token is stored so subsequent commands can be run without including these options. This token remains valid for five minutes after the last command that used it.

- `h`, `--help`
  Displays the help for the command.

- `c`, `--use-certificate`
  Use client certificate to sign in. Required when mutual SSL is enabled.
  For information about configuring the certificate, start with the following topic appropriate for your Tableau Server OS:
  - **Windows**: [Configure Mutual SSL](#)
  - **Linux**: [Configure Mutual SSL](#)

- `s`, `--server`
  The Tableau Server URL, which is required at least once to begin session.

- `u`, `--user`
  The Tableau Server username, which is required at least once to begin session.

- `p`, `--password`
  The Tableau Server password, which is required at least once to begin session.
--password-file

Allows the password to be stored in the given .txt file rather than the command line for increased security.

-t, --site

Indicates that the command applies to the site specified by the Tableau Server site ID. If you do not specify a site, the Default site is assumed. Applies only to servers with multiple sites.

-x, --proxy

Host:Port
Uses the specified HTTP proxy.

--no-prompt

When specified, the command will not prompt for a password. If no valid password is provided the command will fail.

--no-proxy

When specified, an HTTP proxy will not be used.

--no-certcheck

When specified, tabcmd (the client) does not validate the server’s SSL certificate.

--[no-]cookie

When specified, the session ID is saved on login so subsequent commands will not need to log in. Use the no- prefix to not save the session ID. By default the session is saved.
**--timeout**

Waits the specified number of seconds for the server to complete processing the command. By default the process will timeout in 30 seconds.

**--**

Specifies the end of options on the command line. You can use -- to indicate to tabcmd that anything that follows -- should not be interpreted as an option setting and can instead be interpreted as a value for the command. This is useful if you need to specify a value in the command that includes a hyphen. The following example shows how you might use -- in a tabcmd command, where -430105/Sheet1 is a required value for the export command.

```
tabcmd export --csv -f "D:\export10.csv" -- -430105/Sheet1
```

**reencryptextracts**

Reencrypt all extracts on a site with new encryption keys. This command will regenerate the key encryption key and data encryption key. If no site is specified, extracts on the default site will be reencrypted. For more information, see Extract Encryption at Rest.

Depending on the number and size of extracts, this operation may consume significant server resources. Consider running this command outside of normal business hours.

**Example**

```
tabcmd reencryptextracts "West Coast Sales"
```

**Global options**

The following options are used by all tabcmd commands. The --server, --user, and --password options are required at least once to begin a session. An authentication token is stored so subsequent commands can be run without including these options. This token remains valid for five minutes after the last command that used it.
-h, --help

Displays the help for the command.

-c, --use-certificate

Use client certificate to sign in. Required when mutual SSL is enabled.

For information about configuring the certificate, start with the following topic appropriate for your Tableau Server OS:

- Windows: Configure Mutual SSL
- Linux: Configure Mutual SSL

-s, --server

The Tableau Server URL, which is required at least once to begin session.

-u, --user

The Tableau Server username, which is required at least once to begin session.

-p, --password

The Tableau Server password, which is required at least once to begin session.

--password-file

Allows the password to be stored in the given .txt file rather than the command line for increased security.

-t, --site

Indicates that the command applies to the site specified by the Tableau Server site ID.
If you do not specify a site, the Default site is assumed. Applies only to servers with
Tableau Server on Linux Administrator Guide

multiple sites.

-x, --proxy

Host:Port

Uses the specified HTTP proxy.

--no-prompt

When specified, the command will not prompt for a password. If no valid password is provided the command will fail.

--no-proxy

When specified, an HTTP proxy will not be used.

--no-certcheck

When specified, tabcmd (the client) does not validate the server's SSL certificate.

-[no-]cookie

When specified, the session ID is saved on login so subsequent commands will not need to log in. Use the no- prefix to not save the session ID. By default the session is saved.

--timeout

Waits the specified number of seconds for the server to complete processing the command. By default the process will timeout in 30 seconds.

--

Specifies the end of options on the command line. You can use -- to indicate to tabcmd that anything that follows -- should not be interpreted as an option setting and can
instead be interpreted as a value for the command. This is useful if you need to specify a value in the command that includes a hyphen. The following example shows how you might use -- in a tabcmd command, where -430105/Sheet1 is a required value for the export command.

```
tabcmd export --csv -f "D:\export10.csv" -- -430105/Sheet1
```

**refreshextracts workbook-name or datasource-name**

Performs a full or incremental refresh of extracts belonging to the specified workbook or data source.

This command takes the name of the workbook or data source as it appears on the server, not the file name when it was published. Only an administrator or the owner of the workbook or data source is allowed to perform this operation.

**Examples**

```
tabcmd refreshextracts --datasource sales_ds
```

```
tabcmd refreshextracts --project "Sales External" --datasource sales_ds
```

```
tabcmd refreshextracts --workbook "My Workbook"
```

```
tabcmd refreshextracts --url SalesAnalysis
```

```
tabcmd refreshextracts --workbook "My Workbook" --addcalculations
```

```
tabcmd refreshextracts --datasource sales_ds --removecalculations
```

**Options**

```
--incremental
```

Runs the incremental refresh operation.
--synchronous

Adds the full refresh operation to the queue used by the Backgrounder process, to be run as soon as a Backgrounder process is available. If a Backgrounder process is available, the operation is run immediately. The refresh operation appears on the Background Tasks report.

During a synchronous refresh, `tabcmd` maintains a live connection to the server while the refresh operation is underway, polling every second until the background job is done.

--workbook

The name of the workbook containing extracts to refresh. If the workbook has spaces in its name, enclose it in quotes.

--datasource

The name of the data source containing extracts to refresh.

--project

Use with --workbook or --datasource to identify a workbook or data source in a project other than Default. If not specified, the Default project is assumed.

--parent-project-path

Specifies the name of the parent project for the nested project as specified with the --project option. For example, to specify a project called "Nested" that exists in a "Main" project, use the following syntax: --parent-project-path "Main" --project "Nested".

--url

The name of the workbook as it appears in the URL. A workbook published as "Sales
Analysis” has a URL name of “SalesAnalysis”.

**--addcalculations**

Use with **--workbook** to materialize calculations in the embedded extract of the workbook or **--datasource** to materialize calculations in the extract data source. Adds the operation to the queue used by the Backgrounder process. If a Backgrounder process is available, the operation runs immediately. This operation appears on the **Background Tasks for Extracts** administrative view.

**--removecalculations**

Use with **--workbook** or **--datasource** to remove calculations that were previously materialized. Adds the operation to the queue used by the Backgrounder process. If a Backgrounder process is available, the operation runs immediately. This operation appears on the **Background Tasks for Extracts** administrative view.

**Global options**

The following options are used by all **tabcmd** commands. The **--server**, **--user**, and **--password** options are required at least once to begin a session. An authentication token is stored so subsequent commands can be run without including these options. This token remains valid for five minutes after the last command that used it.

**-h, --help**

Displays the help for the command.

**-c, --use-certificate**

Use client certificate to sign in. Required when mutual SSL is enabled.
For information about configuring the certificate, start with the following topic appropriate for your Tableau Server OS:

- **Windows**: Configure Mutual SSL
- **Linux**: Configure Mutual SSL

-s, --server

The Tableau Server URL, which is required at least once to begin session.

-u, --user

The Tableau Server username, which is required at least once to begin session.

-p, --password

The Tableau Server password, which is required at least once to begin session.

--password-file

Allows the password to be stored in the given .txt file rather than the command line for increased security.

-t, --site

Indicates that the command applies to the site specified by the Tableau Server site ID. If you do not specify a site, the Default site is assumed. Applies only to servers with multiple sites.

-x, --proxy

Host:Port

Uses the specified HTTP proxy.
--no-prompt

When specified, the command will not prompt for a password. If no valid password is provided the command will fail.

--no-proxy

When specified, an HTTP proxy will not be used.

--no-certcheck

When specified, tabcmd (the client) does not validate the server’s SSL certificate.

--[no-]cookie

When specified, the session ID is saved on login so subsequent commands will not need to log in. Use the no- prefix to not save the session ID. By default the session is saved.

--timeout

Waits the specified number of seconds for the server to complete processing the command. By default the process will timeout in 30 seconds.

--

Specifies the end of options on the command line. You can use -- to indicate to tabcmd that anything that follows -- should not be interpreted as an option setting and can instead be interpreted as a value for the command. This is useful if you need to specify a value in the command that includes a hyphen. The following example shows how you might use -- in a tabcmd command, where -430105/Sheet1 is a required value for the export command.

tabcmd export --csv -f "D:\export10.csv" -- -430105/Sheet1
reset_openid_sub

Clears OpenID Connect identifiers (sub values) that have already been associated with Tableau Server identities. See Changing IdPs in Tableau Server for OpenID Connect.

Example

tabcmd reset_openid_sub --username jsmith

Options

--username

Clears sub value for the specified individual user.

--all

Clears sub values for all users.

Global options

The following options are used by all tabcmd commands. The --server, --user, and --password options are required at least once to begin a session. An authentication token is stored so subsequent commands can be run without including these options. This token remains valid for five minutes after the last command that used it.

-h, --help

Displays the help for the command.

-c, --use-certificate

Use client certificate to sign in. Required when mutual SSL is enabled.
For information about configuring the certificate, start with the following topic appropriate for your Tableau Server OS:

- **Windows**: Configure Mutual SSL
- **Linux**: Configure Mutual SSL

**-s, --server**

The Tableau Server URL, which is required at least once to begin session.

**-u, --user**

The Tableau Server username, which is required at least once to begin session.

**-p, --password**

The Tableau Server password, which is required at least once to begin session.

**--password-file**

Allows the password to be stored in the given .txt file rather than the command line for increased security.

**-t, --site**

Indicates that the command applies to the site specified by the Tableau Server site ID. If you do not specify a site, the Default site is assumed. Applies only to servers with multiple sites.

**-x, --proxy**

Host:Port

Uses the specified HTTP proxy.
Tableau Server on Linux Administrator Guide

--no-prompt

When specified, the command will not prompt for a password. If no valid password is provided the command will fail.

--no-proxy

When specified, an HTTP proxy will not be used.

--no-certcheck

When specified, tabcmd (the client) does not validate the server’s SSL certificate.

--[no-]cookie

When specified, the session ID is saved on login so subsequent commands will not need to log in. Use the no- prefix to not save the session ID. By default the session is saved.

--timeout

Waits the specified number of seconds for the server to complete processing the command. By default the process will timeout in 30 seconds.

--

Specifies the end of options on the command line. You can use -- to indicate to tabcmd that anything that follows -- should not be interpreted as an option setting and can instead be interpreted as a value for the command. This is useful if you need to specify a value in the command that includes a hyphen. The following example shows how you might use -- in a tabcmd command, where -430105/Sheet1 is a required value for the export command.

```
tabcmd export --csv -f "D:\export10.csv" -- -430105/Sheet1
```

removeusers group-name

Removes users from the specified group.
Example

tabcmd removeusers "Development" --users "users.csv"

Options

--users

Remove the users in the given .csv file from the specified group. The file should be a simple list with one user name per line.

--[no-]complete

Requires that all rows be valid for any change to succeed. If not specified --complete is used.

Global options

The following options are used by all tabcmd commands. The --server, --user, and --password options are required at least once to begin a session. An authentication token is stored so subsequent commands can be run without including these options. This token remains valid for five minutes after the last command that used it.

-h, --help

Displays the help for the command.

-c, --use-certificate

Use client certificate to sign in. Required when mutual SSL is enabled.

For information about configuring the certificate, start with the following topic appropriate for your Tableau Server OS:
Tableau Server on Linux Administrator Guide

- **Windows**: Configure Mutual SSL

- **Linux**: Configure Mutual SSL

- `s`, `--server`

  The Tableau Server URL, which is required at least once to begin session.

- `u`, `--user`

  The Tableau Server username, which is required at least once to begin session.

- `p`, `--password`

  The Tableau Server password, which is required at least once to begin session.

- `--password-file`

  Allows the password to be stored in the given `.txt` file rather than the command line for increased security.

- `t`, `--site`

  Indicates that the command applies to the site specified by the Tableau Server site ID. If you do not specify a site, the Default site is assumed. Applies only to servers with multiple sites.

- `x`, `--proxy`

  Host:Port

  Uses the specified HTTP proxy.

- `--no-prompt`

  When specified, the command will not prompt for a password. If no valid password is
provided the command will fail.

--no-proxy

When specified, an HTTP proxy will not be used.

--no-certcheck

When specified, tabcmd (the client) does not validate the server's SSL certificate.

--[no-]cookie

When specified, the session ID is saved on login so subsequent commands will not need to log in. Use the no- prefix to not save the session ID. By default the session is saved.

--timeout

Waits the specified number of seconds for the server to complete processing the command. By default the process will timeout in 30 seconds.

--

Specifies the end of options on the command line. You can use -- to indicate to tabcmd that anything that follows -- should not be interpreted as an option setting and can instead be interpreted as a value for the command. This is useful if you need to specify a value in the command that includes a hyphen. The following example shows how you might use -- in a tabcmd command, where -430105/Sheet1 is a required value for the export command.

    tabcmd export --csv -f "D:\export10.csv" -- -430105/Sheet1

runschedule schedule-name

Runs the specified schedule.

This command takes the name of the schedule as it is on the server.
For Tableau Online, the command can be run within the scope of a single site, using site administrator permissions.

Example

tabcmd runschedule "5AM Sales Refresh"

Global options

The following options are used by all tabcmd commands. The --server, --user, and --password options are required at least once to begin a session. An authentication token is stored so subsequent commands can be run without including these options. This token remains valid for five minutes after the last command that used it.

-h, --help

Displays the help for the command.

-c, --use-certificate

Use client certificate to sign in. Required when mutual SSL is enabled.

For information about configuring the certificate, start with the following topic appropriate for your Tableau Server OS:

- **Windows**: Configure Mutual SSL
- **Linux**: Configure Mutual SSL

-s, --server

The Tableau Server URL, which is required at least once to begin session.

-u, --user

The Tableau Server username, which is required at least once to begin session.
-p, --password

The Tableau Server password, which is required at least once to begin session.

--password-file

Allows the password to be stored in the given .txt file rather than the command line for increased security.

-t, --site

Indicates that the command applies to the site specified by the Tableau Server site ID. If you do not specify a site, the Default site is assumed. Applies only to servers with multiple sites.

-x, --proxy

Host:Port

Uses the specified HTTP proxy.

--no-prompt

When specified, the command will not prompt for a password. If no valid password is provided the command will fail.

--no-proxy

When specified, an HTTP proxy will not be used.

--no-certcheck

When specified, tabcmd (the client) does not validate the server’s SSL certificate.

--[no-]cookie
When specified, the session ID is saved on login so subsequent commands will not need to log in. Use the no- prefix to not save the session ID. By default the session is saved.

--timeout

Waits the specified number of seconds for the server to complete processing the command. By default the process will timeout in 30 seconds.

--

Specifies the end of options on the command line. You can use -- to indicate to tabcmd that anything that follows -- should not be interpreted as an option setting and can instead be interpreted as a value for the command. This is useful if you need to specify a value in the command that includes a hyphen. The following example shows how you might use -- in a tabcmd command, where -430105/Sheet1 is a required value for the export command.

```
tabcmd export --csv -f "D:\export10.csv" -- -430105/Sheet1
```

**set setting**

Enables the specified setting on the server. Details about each setting can be seen on the Maintenance page on the server.

Use an exclamation mark in front of the setting name to disable the setting. You can enable or disable the following settings:

- allow_scheduling
- embedded_credentials
- remember_passwords_forever

**Example**

```
tabcmd set embedded_credentials
```
Global options

The following options are used by all tabcmd commands. The --server, --user, and --password options are required at least once to begin a session. An authentication token is stored so subsequent commands can be run without including these options. This token remains valid for five minutes after the last command that used it.

-h, --help

Displays the help for the command.

-c, --use-certificate

Use client certificate to sign in. Required when mutual SSL is enabled.

For information about configuring the certificate, start with the following topic appropriate for your Tableau Server OS:

- **Windows:** Configure Mutual SSL
- **Linux:** Configure Mutual SSL

-s, --server

The Tableau Server URL, which is required at least once to begin session.

-u, --user

The Tableau Server username, which is required at least once to begin session.

-p, --password

The Tableau Server password, which is required at least once to begin session.

--password-file
Allows the password to be stored in the given .txt file rather than the command line for increased security.

-t, --site

Indicates that the command applies to the site specified by the Tableau Server site ID. If you do not specify a site, the Default site is assumed. Applies only to servers with multiple sites.

-x, --proxy

Host:Port

Uses the specified HTTP proxy.

--no-prompt

When specified, the command will not prompt for a password. If no valid password is provided the command will fail.

--no-proxy

When specified, an HTTP proxy will not be used.

--no-certcheck

When specified, tabcmd (the client) does not validate the server’s SSL certificate.

--[no-]cookie

When specified, the session ID is saved on login so subsequent commands will not need to log in. Use the no- prefix to not save the session ID. By default the session is saved.

--timeout
Waits the specified number of seconds for the server to complete processing the command. By default the process will timeout in 30 seconds.

--

Specifies the end of options on the command line. You can use -- to indicate to tabcmd that anything that follows -- should not be interpreted as an option setting and can instead be interpreted as a value for the command. This is useful if you need to specify a value in the command that includes a hyphen. The following example shows how you might use -- in a tabcmd command, where -430105/Sheet1 is a required value for the export command.

```
tabcmd export --csv -f "D:\export10.csv" -- -430105/Sheet1
```

**syncgroup** *group-name*

Synchronizes a Tableau Server group with an Active Directory group. If the Tableau Server group does not already exist, it is created and synchronized with the specified Active Directory group.

If the group name itself includes an "@" (other than as the domain separator) you need to refer to the symbol using the hex format "\0x40".

**Example**

```
tabcmd syncgroup "Development"
```

```
tabcmd syncgroup "Dev\0x40West"
```

**Note:** If you synchronize a group that you are a member of, changes that you make using this command do not apply to your user. For example, if you use this command to remove the administrator right from users in a group that you are a member of, you are still an administrator when the command finishes.
Options

--no-publisher

Deprecated. Use the --role option instead.

--overwritesiterole

Allows a user’s site role to be overwritten with a less privileged one when using --role. By default, a user site role can be promoted when using --role, but cannot be demoted. Because the --overwritesiterole option will demote user site roles, use it with caution.

--publisher

Deprecated. Use the --role option instead.

-r, --role

Specifies a site role for users in the group. The default is Unlicensed.

--silent-progress

Do not display progress messages for the command.

Global options

The following options are used by all tabcmd commands. The --server, --user, and --password options are required at least once to begin a session. An authentication token is stored so subsequent commands can be run without including these options. This token remains valid for five minutes after the last command that used it.

-h, --help
Displays the help for the command.

- **c, --use-certificate**

  Use client certificate to sign in. Required when mutual SSL is enabled.

  For information about configuring the certificate, start with the following topic appropriate for your Tableau Server OS:
  - **Windows**: Configure Mutual SSL
  - **Linux**: Configure Mutual SSL

- **s, --server**

  The Tableau Server URL, which is required at least once to begin session.

- **u, --user**

  The Tableau Server username, which is required at least once to begin session.

- **p, --password**

  The Tableau Server password, which is required at least once to begin session.

- **password-file**

  Allows the password to be stored in the given .txt file rather than the command line for increased security.

- **t, --site**

  Indicates that the command applies to the site specified by the Tableau Server site ID. If you do not specify a site, the Default site is assumed. Applies only to servers with multiple sites.
-x, --proxy

Host:Port

Uses the specified HTTP proxy.

--no-prompt

When specified, the command will not prompt for a password. If no valid password is provided the command will fail.

--no-proxy

When specified, an HTTP proxy will not be used.

--no-certcheck

When specified, tabcmd (the client) does not validate the server’s SSL certificate.

--[no-]cookie

When specified, the session ID is saved on login so subsequent commands will not need to log in. Use the no- prefix to not save the session ID. By default the session is saved.

--timeout

Waits the specified number of seconds for the server to complete processing the command. By default the process will timeout in 30 seconds.

--

Specifies the end of options on the command line. You can use -- to indicate to tabcmd that anything that follows -- should not be interpreted as an option setting and can instead be interpreted as a value for the command. This is useful if you need to specify a value in the command that includes a hyphen. The following example shows how you
might use -- in a `tabcmd` command, where `-430105/SHEET1` is a required value for the `export` command.

```
tabcmd export --csv -f "D:\export10.csv" -- -430105/SHEET1
```

**version**

Displays the version information for the current installation of the `tabcmd` utility.

**Example**

```
tabcmd version
```

**Global options**

The following options are used by all `tabcmd` commands. The `--server`, `--user`, and `--password` options are required at least once to begin a session. An authentication token is stored so subsequent commands can be run without including these options. This token remains valid for five minutes after the last command that used it.

`-h, --help`

Displays the help for the command.

`-c, --use-certificate`

Use client certificate to sign in. Required when mutual SSL is enabled.

For information about configuring the certificate, start with the following topic appropriate for your Tableau Server OS:

- **Windows:** Configure Mutual SSL
- **Linux:** Configure Mutual SSL

`-s, --server`
The Tableau Server URL, which is required at least once to begin session.

-u, --user

The Tableau Server username, which is required at least once to begin session.

-p, --password

The Tableau Server password, which is required at least once to begin session.

--password-file

Allows the password to be stored in the given .txt file rather than the command line for increased security.

-t, --site

Indicates that the command applies to the site specified by the Tableau Server site ID. If you do not specify a site, the Default site is assumed. Applies only to servers with multiple sites.

-x, --proxy

Host:Port

Uses the specified HTTP proxy.

--no-prompt

When specified, the command will not prompt for a password. If no valid password is provided the command will fail.

--no-proxy

When specified, an HTTP proxy will not be used.
---no-certcheck

When specified, tabcmd (the client) does not validate the server's SSL certificate.

---[no-]cookie

When specified, the session ID is saved on login so subsequent commands will not need to log in. Use the no- prefix to not save the session ID. By default the session is saved.

---timeout

Waits the specified number of seconds for the server to complete processing the command. By default the process will timeout in 30 seconds.

---

Specifies the end of options on the command line. You can use -- to indicate to tabcmd that anything that follows -- should not be interpreted as an option setting and can instead be interpreted as a value for the command. This is useful if you need to specify a value in the command that includes a hyphen. The following example shows how you might use -- in a tabcmd command, where -430105/SHEET1 is a required value for the export command.

```
tabcmd export --csv -f "D:\export10.csv" -- -430105/SHEET1
```

Install Switches and Properties for tabcmd

You can use the following switches when installing the Tableau Server Command Line Utility (tabcmd) version 2019.4.0 or later from the command line on Windows.

<table>
<thead>
<tr>
<th>Switch</th>
<th>Description</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>/install</td>
<td>Run Setup</td>
<td>Default is to install, displaying UI and all prompts. If no directory is specified on a fresh install, C:\Program</td>
</tr>
<tr>
<td>Command</td>
<td>Description</td>
<td>Note</td>
</tr>
<tr>
<td>-------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>----------------------------------------------------------------------</td>
</tr>
<tr>
<td>/uninstall</td>
<td>to either install, repair, or uninstall tabcmd, or with /layout, create a complete local copy of the installation bundle in the directory specified.</td>
<td></td>
</tr>
<tr>
<td>/passive</td>
<td>Run Setup with minimal UI and no prompts.</td>
<td></td>
</tr>
<tr>
<td>/quiet</td>
<td>Run Setup in unattended, fully silent mode. No UI or prompts are displayed.</td>
<td><strong>Note</strong>: Use either /silent or /quiet, not both.</td>
</tr>
<tr>
<td>/norestart</td>
<td>Run Setup without restarting Windows,</td>
<td><strong>Note</strong>: In certain rare cases, a restart cannot be suppressed, even when this option is used. This is</td>
</tr>
</tbody>
</table>
even if a restart is necessary. most likely when an earlier system restart was skipped, for example, during installation of other software.

```
/log "<log-file>"
```

Log information to the specified file and path. If no file location is specified, the log file is written to the user's TEMP folder (C:\Users\<username>\AppData\Local\Temp). Check this log file for errors after installation.

By default log files are created in the user's %TEMP% folder with a naming convention of Tableau_Server_Command_Line_utility_<version_code>.log.

Example:<Setup file> /silent /log="C:\Tableau\Logs\tabcmd-Install" ACCEPTEULA=1

<table>
<thead>
<tr>
<th>Properties</th>
<th>Description</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Parameter</td>
<td>Description</td>
<td>Example</td>
</tr>
<tr>
<td>---------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>---------</td>
</tr>
<tr>
<td>ACCEPTEULA=1</td>
<td>Accept the End User License Agreement (EULA). Required for quiet, silent, and passive install. 1 = accept the EULA, 0 = do not accept the EULA.</td>
<td></td>
</tr>
<tr>
<td>INSTALLDIR=</td>
<td>Install tabcmd to the specified non-default install location.</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Specifies the location to install tabcmd. If not used, tabcmd is installed to C:\Program Files\Tableau\Tableau Server&lt;version_code&gt;\extras\Command Line Utility</td>
<td></td>
</tr>
</tbody>
</table>
Troubleshooting

You can use the following topics to troubleshoot and resolve issues with Tableau Server.

Troubleshoot Tableau Server on Linux

Follow the suggestions in this topic to resolve common issues with Tableau Server. For additional troubleshooting steps based on process status viewed on the Status page, see Troubleshoot Server Processes.

Task 931788: 2019.3 - New content request (Linux): Debian support

The following table displays the default locations of the installation, data, logs, and scripts directories:

<table>
<thead>
<tr>
<th>Directory</th>
<th>Default location</th>
</tr>
</thead>
<tbody>
<tr>
<td>Installation</td>
<td>/opt/tableau/tableau_server</td>
</tr>
<tr>
<td>Data</td>
<td>/var/opt/tableau/tableau_server/data</td>
</tr>
<tr>
<td>Logs</td>
<td>/var/opt/tableau/tableau_server/data/tabsvc/logs/</td>
</tr>
<tr>
<td>Scripts</td>
<td>/opt/tableau/tableau_server-/packages/scripts.&lt;version_code&gt;/</td>
</tr>
</tbody>
</table>

General Troubleshooting Steps

Many Tableau Server issues can be addressed or tested with one or more of these basic steps:
Tableau Server on Linux Administrator Guide

Clean install

Install Tableau Server on Linux on a computer that has never had Tableau installed on it. If you are reusing a computer or VM that has had a previous version of Tableau Server installed, follow the steps in Remove Tableau Server from Your Computer to clean Tableau off your computer before you install the new version.

If you run into problems installing Tableau Server you may need to entirely remove Tableau from your computer and do a clean install. See Remove Tableau Server from Your Computer for details.

Disk space

Make sure there is enough disk space on each computer running Tableau Server. Limited disk space can cause a failure to install, a failure to upgrade, or problems running Tableau Server.

Remove old log files

If you are running out of disk space you can clean up old Tableau Server log files. These can take up space and as a best practice you should remove them regularly.

**Version 10.5.x**

If you have version 10.5.1 and higher, run this command at a terminal prompt to clean up log files you do not need:

```
tsm maintenance cleanup
```

**Version 10.5.0**

If you are running version 10.5.0 of Tableau Server on Linux, the cleanup command is not available and you need to run these commands at a terminal prompt:

```
sudo find /var/opt/tableau/tableau_server/data/tabsvc/temp/* -mtime +2 -type f -delete
```

```
sudo find /var/opt/tableau/tableau_server/data/tabsvc/logs/* -
mtime +2 -type f -delete

**Important:** The Linux file system makes it possible to delete files that are open and if you do this the Tableau processes may not be able to recreate the files. This will result in empty log files. To fix this you can stop Tableau Server, restart the TSM Controller, and restart Tableau again:

1. **Stop Tableau Server:**
   ```
   tsm stop
   ```

2. **Restart the TSM Controller:**
   ```
   sudo systemctl restart tabadmincontroller_0.service
   ```

3. **Wait several minutes for the controller to restart.** You can confirm the controller has restarted with this command:
   ```
   tsm status -v
   ```
   When you can run that command and the Tableau Server Administration Controller is listed as 'running' the controller has restarted.

4. **Start Tableau Server:**
   ```
   tsm start
   ```

**Manually gather logs**

If you cannot run `tsm maintenance ziplogs` for any reason (for example, if you have a critical failure before you run `tsm initialize`), you can manually collect and zip the logs by running these commands in a terminal window:

```
cd /var/opt/tableau/tableau_server/data/tabsvc/

cp /var/opt/tableau/tableau_server/logs/app-install.log logs

cp ~/.tableau/tsm/tsm.log logs
```
tar -czvf ~/logs.tar.gz logs

This creates a file called logs.tar.gz in your home directory. You can upload or send this file to Tableau.

Restart server

Restart Tableau Server. Issues related to indexing and processes not fully started can be resolved by restarting Tableau Server in a controlled way. To restart Tableau Server, use the `tsm restart` command. This will stop all the processes associated with Tableau Server and then restart them.

Edit installation and configuration files using Linux

You should edit or create any files used to install or configure Tableau Server on Linux using a Linux operating system. Files created using Microsoft Windows will cause errors in Tableau Server on Linux installation and configuration because Linux operating systems end files with a line-feed (LF) character, whereas Windows ends files with a carriage-return character and a line-feed character (CR LF). Non-Linux (CR LF) file endings can cause errors during Automated Installation of Tableau Server if they appear in the `config.json`, `reg_template.json` or `secrets` files used by the automated installer. Non-Linux (CR LF) file endings can also cause errors during registration or when configuring identity store settings or gateway settings.

Check systemd logs

If Tableau Server will not start, and you do not find anything useful in the Tableau logs (see Work with Log Files for more information) you can check the logs at `/var/-logs/messages` for messages related to the TSM Service starting and stopping.

Installing Tableau Server

Install fails due to hardware requirements

Tableau Server cannot install if the computer you are installing on does not meet the minimum hardware requirements. For details on requirements, see Before you install....
Install fails due to timeouts

If you install Tableau Server on a computer with limited resources, for example, a computer that just meets the minimum hardware requirements, you can run into problems where tsm commands timeout due to slow response. You can specify a longer timeout by using the global --request-timeout option on all tsm commands. For more information on the --request-timeout option, see for example, tsm initialize.

Install fails with "Failed to initialize the instance of the temporary database"

Tableau Server on Linux only supports UTF-8 character encoding. If your Linux locale is missing the UTF-8 encoding, your installation can fail with an error similar to this one:

Failed to initialize the instance of the temporary database

To check if your locale is using UTF-8 encoding, run the localectl command at a command prompt. The resulting output should look something like this (your locale may be different):

```
[tableauserver-centos1a ~]$ localectl
System Locale: LANG=en_US.UTF-8
[tableauserver-centos1a ~]$>
```

If the LANG value does not include .UTF-8 then you need to run localectl to add it:

```
sudo localectl set-locale LANG=<your_locale>.UTF-8
```

**Note:** In some cases localectl may not complete (timeout) if your version of systemd is old. Updating systemd may fix this problem and allow you to set the UTF-8 encoding. On RHEL-like systems, use this command to update systemd:

```
sudo yum update systemd
```

Installation fails on a virtual machine in Parallels

Parallels is currently not supported. If you install Tableau Server on a Linux virtual machine in Parallels, the install might fail.
Tableau Server doesn't start

If Tableau Server does not start or is running in a degraded state, run the `tsm restart` command. This will shut down any processes that are running, and restart Tableau Server.

Cannot start Tableau Server after installation

Tableau Server might not start if your computer's hostname changes after installation. One of the main reasons why the hostname might change is if you use the `cloud-init` package on CentOS. If you use the cloud-init package, *reboot the computer* where you want to install Tableau Server before you begin the installation process. Alternatively, you can fix the hostname without rebooting by running the following command:

```
sudo hostnamectl set-hostname `hostnamectl --static`
```

The cloud-init package is commonly used to initialize new virtual machines, configure SSH public key authentication, and more. For example, some CentOS images use cloud-init, and cloud-init is commonly used in OpenStack deployments. However, the version of cloud-init included by default in the CentOS 7.x repositories (cloud-init 0.7.5-10.el7.centos.1) has a known issue that prevents your computer from displaying its Fully Qualified Domain Name (FQDN) along with its hostname until after it restarts.

Because the Tableau Server installation process uses your computer's hostname to configure server processes and generate TLS certificates, Tableau Server might not start if it is configured to use a hostname without the FQDN.

To determine if your computer is displaying the correct hostname, run the `hostnamectl` command. In the following example, the command displays a transient hostname which indicates that it will not return the FQDN and must be restarted.

```
$ hostnamectl
   Static hostname: server01.example.com
Transitory hostname: server01
[...]
```
Alternatively, in the following example, the command displays the correct hostname and FQDN:

```bash
$ hostnamectl
   Static hostname: server01.example.com
[...]
```

Cannot create initial administrator account with multiple Active Directory (AD) domains

When you create the initial administrator account on Tableau Server, you might see the following error if you selected AD as the authentication type:

Failed to authenticate username and password

This occurs when Tableau Server attempts to connect to multiple AD domains. For example, you might see this error if you install Tableau Server on a computer that is part of one domain and you attempt to authenticate AD users that are part of another domain.

Fonts

Tableau Server uses the fonts installed on the system to render workbooks based on the fonts used when a workbook was created. When a font is not available, Tableau Server will use the closest equivalent based on font families; this is true for both Windows and Linux Servers. On Linux Servers missing fonts may be more obvious because Linux ships with fewer included fonts than Windows and OS/X systems do. This matters because many workbooks are authored in Tableau Desktop on Windows or on Mac.

Tableau Server on Linux ships with the following fonts:

- Arial
- Courier
- Georgia
- Times New Roman
- Verdana
- Trebuchet MS
- Tableau Font
Tableau Server on Linux Administrator Guide

Workbooks which use fonts other than these may appear differently than expected when viewed on Tableau Server on Linux, due to missing fonts. To resolve this issue, install the appropriate fonts onto all nodes in your Tableau Server installation.

Support for Asian character sets

If you see empty boxes where you expect to see Asian characters in workbooks that are displayed on Tableau Server, then you should install the language-appropriate font packages in your Linux environment.

Initializing Tableau Server

TSM initialization fails because the `tableau` user account exists but is not a member of the group `tableau`

When you install and initialize Tableau Services Manager (TSM) and Tableau Server, the initialization script (`initialize-tsm`) creates the users and groups needed to run, or confirms that the existing ones are configured with the required characteristics. By default the script creates a user called `tableau` and adds it to a group called `tableau`. If a `tableau` user already exists but is not part of the `tableau` group, the script fails with a warning.

If this happens you can fix the conflict by using a `--unprivileged-user` flag to specify a different user, and the user will be created and added to the `tableau` group.

For example, to specify a user named `tableauserver`, you would run the script from the `/opt/tableau/tableau_server/packages/scripts.<version_code>` directory using this command:

```
sudo ./initialize-tsm --unprivileged-user="tableauserver" --accepteula
```

For a complete list of options that can be used with the `initialize-tsm` script, use the `-h` option:

```
sudo ./initialize-tsm -h
```
Error initializing Tableau Server on unsupported system locale

If you attempt to install Tableau Server on a computer with a locale that is not one of the 11 supported locales, you will get an error during installation.

Tableau Server will run on a system using one of the following locales:

de_DE, en_GB, en_US, es_ES, fr_FR, it_IT, ja_JP, ko_KR, pt_BR, zh_CN, or zh_TW.

Any other locale will generate the error.

Error initializing Tableau Server when en_US.utf8 is not included in locale list

If you attempt to install Tableau Server on a computer that does not have en_US.utf8 in the locale list, the initialization will fail with an error. To see if en_US.utf8 is included, type locale -a at a shell prompt.

If en_US.utf8 is not listed, you can en_us to the locale list by typing sudo locale-gen en_US.UTF-8 at a shell prompt on Ubuntu and Debian, or sudo localedef -i en_US -f UTF-8 at a shell prompt on RHEL-like distributions.

Error: status 10 - initializing Tableau Server when data directory path includes a period

If you attempt to install Tableau Server and specify a data directory with a path that includes a period ("."), initialization will fail with errors including:

Connection timed out

and

ERROR: TSM services returned status 10

To avoid this issue, choose a data directory that does not include a period in its path.

Error initializing Tableau Server after reinstallation

If you uninstall and reinstall Tableau Server, you can encounter an error initializing Tableau Server. For example, you might see the following error:
Tableau Server on Linux Administrator Guide

**ERROR** com.tableau.tabadmin.webapp.asyncjobs.JobStepRunner - Running step WaitForConfigure failed
com.tableau.tabadmin.webapp.exceptions.ServiceFailedStateException

This error occurs when artifacts remain from a previous installation that cause services to fail to start. To prevent this error, use the `tableau-server-oblit`er`ate` script in the `/opt/tableau/tableau_server/packages/scripts.<version_code>` folder. For more information about completely removing Tableau Server, see Remove Tableau Server from Your Computer.

**Activating Tableau Server**

Tableau Server license activation fails

In certain cases activation of the Tableau product key using the `tsm licenses activate -k <product_key>` command fails with an error:

License Server not available

This can happen if your computer is unable to connect through TCP port 443 to the Tableau licensing server at `licensing.tableau.com`.

To resolve this you need to configure your network and/or host-based firewalls to allow access to that address and port, or activate Tableau offline. For more information, see Activate Tableau Server Offline - Adding a License.

**Reindexing Tableau Server Search & Browse**

Problems that can be solved by reindexing Search & Browse

Symptoms of an index that needs to be rebuilt include:

- A blank list of sites when a user attempts to log in
- A blank list of projects when a user tries to select a project
- Missing content (workbooks, views, dashboards)
- Unexpected or inaccurate alerts (for example, an "refresh failed" alert on a workbook that does not include an extract)
If you see any of these behaviors, rebuild the Search & Browse index using the `tsm maintenance reindex-search` command.

**Restarting Tableau Server**

**Restarting Tableau Server or applying changes fails**

If one of the Tableau Server services fails, you might see an error when you attempt to restart the server or apply configuration changes.

To see if a failed service is causing the error, type the following command:

```bash
tsm status -v
```

To find out why a service failed, view the `tabadminagent` and `tabadmincontroller` log files in the data directory. For example, a service might fail because of concurrency issues or port configuration issues. Please include any issues you encounter in your feedback.

As a workaround, you can attempt to resolve the failure by removing and re-adding the service in TSM. Once the service has started, you can retry your previous configuration change or retry restarting the server with the `tsm restart` command.

**Error restarting Tableau Server after adding or configuring a node**

If you add a or configure the node without a Gateway process, Tableau Server might fail to restart and you could see errors like these:

```plaintext
ERROR : com.tableau.tabadmin.configuration.PortConfigurationExtractor
Unable to find port config key worker1.gateway.port
```

and

```plaintext
Message: Missing port configuration value for key 'worker1.gateway.port'
```

These errors appear in the `gateway.log` file and occur when a Tableau Server node is configured with either an Application Server or VizQL Server but without a Gateway. A
Gateway process is required if either Application Server or VizQL Server is running on a node.

Backup/Restore

Problems related to restoring a backup created by Tableau Server can be the result of permissions issues. Proper permissions are necessary for both the file that TSM is restoring, and the location of the file. When TSM handles the backup, it puts the file in a default location and sets permissions appropriately. You can run into permission problems if you are restoring a backup that was copied to your Linux server, or a backup from a non-default location on your server. For details about using a non-default location, including how to change the location, see tsm File Paths.

Errors may include:

Server Was Denied Access to File

or

Restoring the backup '<backup>.tsbak' was unsuccessful

or

Comparing authentication methods failed

The Tableau Server backup and restore processes must have:

- Read permission—The processes need to access the .tsbak backup file directly.
- Execute permission—The processes also require execute permissions to the directory structure in which the .tsbak file is placed.

When TSM creates a backup in the default location, it sets the permissions it needs. If you copy a file to the Linux server, or move it to a non-default directory, the permissions may not allow the TSM processes proper access. You need to verify that both the file, and the directory tree that contains it, allow access by the TSM user tableau. The file permissions must give the tableau user read access to the .tsbak file. You can do this by setting the group on
the file to the \textit{tableau} group, and giving that group read access. The directory permissions must give the \textit{tableau} user read access. You can do this by setting the group on the directory to the \textit{tableau} group, and giving that group read and execute access on the directories.

For detailed information about TSM and file permissions, see Files and Permissions in TSM.

\textbf{File locations}\n
Changing basefilepath does not change the location of an existing file

Several tsm commands write files to default locations. You can change these default locations for each command using a tsm set command, but doing so does not move any existing files from the original location to the new one, and it does not create the new location. You are responsible for creating the new location, and for making sure it has the correct permissions to allow tsm access to any files in the location, and the entire directory structure that contains the files.

For more information about changing default locations for backup, restore, site import and export, and ziplogs files, see tsm File Paths.

For information about tsm permissions, see Files and Permissions in TSM.

\textbf{TSM commands}\n
TSM command line does not show progress for long-running tasks

If you run a tsm command such as restore or ziplogs that takes more than 2 hours to complete, the command will continue to run until completion on the server. To display the progress of the job, use the \texttt{tsm jobs reconnect} command.
Opening Firewall ports

Manually opening firewall ports on Ubuntu and Debian

The current version of Tableau Server does not support the `ufw` firewall that is used on Ubuntu and Debian. For customers that do not want to install `firewalld` on Ubuntu and Debian, another option is to manually open those ports. The following steps will confirm that `ufw` is running, and open TCP ports 8850 and 80 to connections from any source address:

1. Run the following command to confirm `ufw` is running:

   ```
sudo ufw status
   ```

   If the result is **Status: inactive**, you will need to enable `ufw` and ensure that you can continue to connect via `ssh`, which is outside the scope of these release notes.

2. Run the following command to allow access to port 8850:

   ```
sudo ufw allow 8850
   ```

3. Run the following command to allow access to port 80:

   ```
sudo ufw allow 80
   ```

OpenID fails on first sign-in attempt

If you have configured Open ID Connect authentication for Tableau Server, the first sign-in attempt fails. To successfully log in, users must retry authentication after the initial failure.

Administrative views do not display

The Status tab of Tableau Server includes links to visualizations that display server metrics. These visualizations require the PostgreSQL driver to access the appropriate data from the Tableau Server repository. The PostgreSQL driver is not installed automatically, so if you have not installed the driver, the views will not display. For more information, see Database Drivers.
Note: To use administrative views, the PostgreSQL driver must be installed on any node that runs the VizQL Server process.

Changing locale on view

When you change your user locale after opening a view, any subsequent attempt to open the view will fail with an "unexpected error." You can still open views that you have not previously opened.

To work around this issue, sign out of Tableau Server after changing your locale, and then sign back in. All views will display properly.

Work with Log Files

Tableau Server creates log files as a normal part of its activities. You may need to use the server log files when you are troubleshooting issues with Tableau Server or if Tableau Support requests logs from you to help you resolve an issue.

You can create a zipped log file archive using the `tsm maintenance ziplogs` command. The zipped archive contains copies of the logs you can unzip and look at, or send to Tableau Support. Once you have a copy of the archive, you can delete the archive from your server. For more information on log file archives, see Log File Snapshots (Archive Logs).

This collection of topics provides information about how to create log file archives, the contents of specific log files, and details about when and how you might want to look at a log.

Investigating Tableau Server Issues

The range and complexity of possible issues with Tableau Server means that there is no simple process you can use to investigate all problems, but a general approach would include these steps:
1. **Clean up** existing log files to reduce their size. For more information, see For more information, see Remove Unneeded Files.

2. **Set the appropriate logging level.** This is something that Tableau Support will instruct you on. For more information, see Change Logging Levels.

3. **Reproduce the issue** you are troubleshooting so the logs capture the events related to the problem.

4. **Create an archive** of the logs. For more information see Log File Snapshots (Archive Logs).

   Important: Use this archive when looking at the log files. You should not edit, move or delete any files directly on the server.

5. **Review the TSM Administration Controller log** (/tabadmincontroller/tabadmincontroller_node<n>-<n>.log) to understand any configuration or deployment done by TSM from the command line, Web UI, or API, including jobs started by TSM. Start with the controller log. This is where you’ll get most useful information.

6. **Review the Apache logs** (/httpd/access.###_##_##_##_##_##.log and /httpd/error.log) for requests that may be related to the issue you are investigating.

   The Apache logs will contain a fair amount of "noise" that does not apply to issues you are experiencing.

   - If you find a request that seems to be related to your issue, search thevizqlserver directory for entries that include the unique request ID from the Apache logs.
   - Look for the response code and message associated with the request ID.
   - Search for the name of the workbook, view, dashboard, or data source that is related to your issue. Make sure to look for a relevant timestamp.
   - If you find a request that seems to be related to your issue, look at the response code associated with the request. (200s are good, 500s indicate problems.)
• Locate the unique request ID associated with the request you’ve identified (the unique request ID is a 24 character alphanumeric string at the very end of the request).

7. Review the log archive further to search for other messages and possible errors.

• Use the request ID from the Apache logs to search the vizqlserver folder of the log archive for files containing related log entries. Look for indications of a problem (for example, error messages or long-running queries).

8. Review script logging.

Tableau Server includes logs for most of the bash scripts that are included in the scripts directory at /opt/tableau/tableau_server-/packages/scripts.<version_code>/. These logs are saved to the /var/tmp directory each time a script is run.

• <install_drive>\<install\path>\logs
by default:
• C:\ProgramData\Tableau\Tableau Server\logs

9. Contact support

If you are not able to solve the issue yourself, or if requested by Tableau Support, send the zipped archive to Tableau.

Server Log File Locations

By default, Tableau Server log file archives are gathered in a zip file called logs.zip, but you can specify a different filename when you create the archive with the tsm maintenance ziplogs command. You can copy the archive from the server to a local computer and open it there, or send it to Tableau Support.

Tableau Server generates log files during normal operations, but these files are especially important when there are problems with Tableau Server. The logs are on a running server, and you can find them, but we strongly recommend that if you are going to be looking at the logs, or if you need to send them to Tableau Support, you use the tsm maintenance ziplogs command to generate an archive of the logs. This command not only gathers
together and zips up the logs, but provides you with a copy you can move to another computer, or send to Tableau.


When you unzip the archive, a series of files are created that start with the directory names shown below, and that end with a version number. These files contain the log files from the corresponding directories. The table in this topic explains the possible contents of these files, along with the original location the files came from on Tableau Server, the process that created the log files, and details about the files.

Tableau Server Log and Configuration File Locations

The Tableau Server log files are found in the following directory: /var/-opt/tableau/tableau_server/data/tabsvc/logs/

Logs not included in the ziplog archive

The tsm maintenance ziplogs command creates an archive of the most important Tableau Server logs, but some logs are not zipped up in the archive. These logs are not included in a log file archive:

- The TSM log. The tsm.log file is located in <home dir>/.tableau/tsm.
- Install and upgrade logs. If your installation or upgrade fails, check these logs first for errors:
  - The install log. The app-install.log file is located in /var/-opt/tableau/tableau_server/logs.
  - The additional node install log. The app-worker-install.log file is located in /var/opt/tableau/tableau_server/logs on additional nodes.
  - The upgrade log. The app-upgrade.log file is located in /var/-opt/tableau/tableau_server/logs.
Bash script logging

Tableau Server includes logs for most of the bash scripts that are included in the scripts directory at `/opt/tableau/tableau_server/packages/scripts.<version>`. These logs are not included in the log file archive. Instead, the logs are saved to the `/var/tmp` directory each time a script is run.

System information

The log archives include system information gathered by running a number of system commands. The output for those commands are included in a subdirectory of the tabadminagent directory called sysinfo. See tabadminagent in the table below for more information.

Log Archive File Locations (unzipped archive)

<table>
<thead>
<tr>
<th>Log Archives in logs.zip</th>
<th>Details</th>
<th>Files</th>
<th>Generated by</th>
<th>Location on Tableau Server</th>
</tr>
</thead>
<tbody>
<tr>
<td>activemqserver</td>
<td>Logs related to the Tableau Server Messaging Service.</td>
<td>control_activemqserver_node&lt;node&gt;-&lt;instance&gt;.log</td>
<td>run-activemqserver_0.cmd</td>
<td>/log-s/activemqserver</td>
</tr>
<tr>
<td></td>
<td></td>
<td>stdout_activemqserver_node&lt;node&gt;-&lt;instance&gt;.yyyy-mm-dd.log activemq.log</td>
<td></td>
<td></td>
</tr>
<tr>
<td>app-zookeeper</td>
<td>Logs related to the Tableau Server</td>
<td>appzookeeper#.log control_app-zookeeper.exe</td>
<td>app-zookeeper.exe</td>
<td>/log-s/appzookeeper</td>
</tr>
<tr>
<td>Service</td>
<td>Logs related to subscriptions and scheduled activities like extract refreshes, &quot;Run Now&quot; tasks, and tabcmd tasks.</td>
<td>/log-s/backgrounder</td>
<td></td>
<td></td>
</tr>
<tr>
<td>------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------</td>
<td>---------------------</td>
<td></td>
<td></td>
</tr>
<tr>
<td>backgrounder</td>
<td>backgrounder-#.log spawn.#####.log run-backgrounder.exe</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>backuprestore</td>
<td>control-backuprestore-#.log</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>cache server</td>
<td>control-cache-server-#.log redis_server.exe</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>cluster controller</td>
<td>clustercontroller.log run-clustercontroller.exe</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>database</td>
<td>control-data-basemaintenance-#.log</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Maintenance</td>
<td>Tenance.</td>
<td>Data Maintenance</td>
<td></td>
<td></td>
</tr>
<tr>
<td>-------------</td>
<td>---------</td>
<td>------------------</td>
<td>---</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>database-#maintenance-.log</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>db-migrate.log</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>migration.log</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>spawn.####.log</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Data-server</td>
<td>Logs related to connections to Tableau Server data sources.</td>
<td>dataserver-#.log</td>
<td>run-data-server.exe</td>
<td>/logs/dataserver</td>
</tr>
<tr>
<td>Elasticserver</td>
<td>Logs related to the Elastic Server process that is used by Ask Data.</td>
<td>control_elasticserver_node&lt;n&gt;-.log</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>elasticsearch.log</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>elasticsearch_deprecation.log</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>elasticsearch_index_indexing_slowlog.log</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>elasticsearch_index_search_slowlog.log</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>stdout_elasticserver_node&lt;n&gt;-.log</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Filestore</td>
<td>Logs related to the Tableau Server File Store process that controls the storage of extracts and syncs extracts across nodes.</td>
<td>filestore.log</td>
<td>run-filestore.exe</td>
<td>/logs/filestore</td>
</tr>
<tr>
<td>-----------</td>
<td>-------------------------------------------------------------------------------------------------</td>
<td>-----------------</td>
<td>-------------------</td>
<td>-----------------</td>
</tr>
<tr>
<td>Flowprocessor</td>
<td>Logs related to Tableau Prep Conductor.</td>
<td>control_flowprocessor_node&lt;n&gt;-&lt;n&gt;.log</td>
<td>run-flowprocessor.exe</td>
<td>/logs/flowprocessor</td>
</tr>
<tr>
<td></td>
<td></td>
<td>flowprocessor_node&lt;n&gt;-&lt;n&gt;.log</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>nativeapi_flowprocessor.txt</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>stdout_flowprocessor_node&lt;n&gt;-&lt;n&gt;.log</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>tdeserver_flowprocessor.txt</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>tomcat_flowprocessor</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>node&lt;node&gt;-&lt;node&gt;.log</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>---</td>
<td>----------------------</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>httpd</strong></td>
<td>Apache logs. Look here for authentication entries. Each request in the Apache log will have a request ID associated with it. This request ID is used throughout the server logs and you can use it to associate log entries with a request.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>access.###-##-##.log</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>error.log</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>startup.log</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Apache daemon</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>/logs/httpd</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>hyper</strong></td>
<td>Logs related to Tableau data engine. A log file is generated each day with information about the Tableau data engine.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>checklicense.log</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>control-hyper-0.log</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>hyper_###-##-##.log</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>/logs/hyper</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Icon</td>
<td>Logs related to</td>
<td>Log file pattern</td>
<td>Path</td>
<td></td>
</tr>
<tr>
<td>------</td>
<td>-----------------</td>
<td>------------------</td>
<td>-------------------</td>
<td></td>
</tr>
</tbody>
</table>
| ![Interactive](interactive-icon.png) | Interactive Microservice Container. | control_interactive_<node>-<instance>.-log  
interactive_<node>-<instance>.log  
stdout_interactive_<node>-<instance>.yyyy-mm-dd.log  
tomcat_non-interactive_<node>-<instance>.yyyy-mm-dd.log | /logs/interactive |
| ![License Service](license-service-icon.png) | Licensing processes. | controllicenseservice.log  
tabcsrv.log | /logs/licenseservice |
<p>| <img src="nlp-icon.png" alt="NLP" /> | Ask Data service | control_nlp_node&lt;n&gt;-&lt;n&gt;.log | /logs/nlp |</p>
<table>
<thead>
<tr>
<th>Logs related to</th>
<th>Description</th>
<th>Path</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>non-interactive</td>
<td>Logs related to the Non-Interactive Microservice Container.</td>
<td>control_non-interactive_&lt;node&gt;-&lt;instance&gt;.log</td>
<td>/log-s/noninteractive</td>
</tr>
<tr>
<td></td>
<td></td>
<td>noninteractive_ &lt;node&gt;-&lt;instance&gt;.log</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>stdout_non-interactive_&lt;node&gt;-&lt;instance&gt;.yyyy-mm-dd.log</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>tomcat_non-interactive_&lt;node&gt;-&lt;instance&gt;.yyyy-mm-dd.log</td>
<td></td>
</tr>
<tr>
<td>psql</td>
<td>Logs related to the PostgreSQL database (repository), including files related to launching server processes</td>
<td>control-psql-#.log</td>
<td>tabspawn</td>
</tr>
<tr>
<td></td>
<td></td>
<td>shut-downCommand.#####.log</td>
<td></td>
</tr>
<tr>
<td>samlser-</td>
<td>Logs related</td>
<td>control-samlser-</td>
<td>/logs/samlservice</td>
</tr>
<tr>
<td>Service</td>
<td>Logs Related to the Secure-SiteAssertion Markup Language (SAML) Service in Tableau Server.</td>
<td>Logs Related to Search Indexing.</td>
<td>Logs Related to Site Import and Export Operations.</td>
</tr>
<tr>
<td>------------------</td>
<td>-----------------------------------------------------------------------------------------</td>
<td>---------------------------------</td>
<td>---------------------------------------------------</td>
</tr>
<tr>
<td>vice</td>
<td>vice-#.log</td>
<td>control-search-server-#.log</td>
<td>control-siteimportexport-0.log</td>
</tr>
<tr>
<td>search-server</td>
<td>Logs related to search indexing.</td>
<td>control-search-server-#.log</td>
<td>control-siteimportexport-0.log</td>
</tr>
<tr>
<td>siteimportexport</td>
<td>Logs related to site import and export operations.</td>
<td>control-siteimportexport-0.log</td>
<td>siteimportexport-0.log</td>
</tr>
<tr>
<td>tabadminagent</td>
<td>Logs related to configuration and topology changes on each server node. Also included in a /sysinfo subdirectory is output from control_tabadminagent_node&lt;#&gt;-&lt;instance#&gt;.log</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Various system commands provide details on the server machine and its state.

<table>
<thead>
<tr>
<th>Log Type</th>
<th>Description</th>
<th>File Path</th>
</tr>
</thead>
<tbody>
<tr>
<td>tabadmin-mincontroller</td>
<td>Logs related to the Tableau Services Manager (TSM) CLI and TSM API.</td>
<td>control-tabadmin-mincontroller-#.log</td>
</tr>
<tr>
<td></td>
<td></td>
<td>tabadmincontroller-#.log</td>
</tr>
<tr>
<td></td>
<td></td>
<td>/logs/tabadmin-mincontroller</td>
</tr>
<tr>
<td>tabsvc</td>
<td>Logs related to the start-up and shut-down of other Tableau Server processes.</td>
<td>control-tabsvc-#.log</td>
</tr>
<tr>
<td></td>
<td></td>
<td>tabsvc.log</td>
</tr>
<tr>
<td></td>
<td></td>
<td>/logs/tabsvc</td>
</tr>
<tr>
<td>vizportal</td>
<td>Logs related to administrative tasks, workbook and permissions management.</td>
<td>vizportal-0.log</td>
</tr>
<tr>
<td></td>
<td></td>
<td>vizportal-0.log.#####-###</td>
</tr>
<tr>
<td></td>
<td></td>
<td>run-vizportal.exe</td>
</tr>
<tr>
<td></td>
<td></td>
<td>/logs/vizportal</td>
</tr>
<tr>
<td>Requests</td>
<td></td>
<td></td>
</tr>
<tr>
<td>------------------</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Authentication,</td>
<td></td>
<td></td>
</tr>
<tr>
<td>sign-ins, initial view requests, and publishing requests.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>These log files also include entries related to REST API requests.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>REST API log entries start with the receipt of the HTTP request, followed by information related to the processing of that request.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>If the request is completed, these log entries are recorded.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Entries end with the HTTP response code for that request.

| VizQL Server | Logs related to displaying and interacting with views. When running multiple instances of VizQL Server, the instances are distinguished by port number. Notify production logs contain exceptional events. | control_vizqlserver_node#-0.log | run-vizqlserver.exe | /logs/vizqlserver |
Log File Snapshots (Archive Logs)

Tableau Server includes functionality to generate a snapshot of log files for archival purposes. If you plan to clean up and delete old log files as part of regular server maintenance, you may consider archiving log files to an off-server storage location before deleting them.

Or, if you are working with Tableau Support on a case, the support engineer may request a server log file snapshot.

This topic describes:

- How to generate a log file snapshot
- How to send the snapshot directly to Tableau Support from the Tableau Server administration tools
- How to download the snapshot
- How to delete archived logs

Use the TSM web interface

1. Open TSM in a browser:

   https://<tsm-computer-name>:8850. For more information, see Sign in to Tableau Services Manager Web UI.

2. Click the **Maintenance**.

3. Generate a log file snapshot.

   a. On the Server Maintenance page, under Log Files, click **Generate Log File Snapshot**.

      An options dialog displays:
b. On the Options page, enter or select the options you want, including a **Description**, time **Range** of log files to be included, and the optional types of logs to be included *(Include Postgres Data, Include Recent Crash Dumps)*, then click **Generate Log File Snapshot**.

The log file snapshot is saved to a fixed location on the computer where TSM and Tableau Server are installed. If you have a multi-node installation, the snapshot is saved to the initial node of the cluster. The location is specified by the `basefilepath_log_archives` variable.

By default the snapshot is saved to:

```
/var/opt/tableau/tableau_server/data/tabsvc/files/log-archives
```

You can find the current location by querying the `basefilepath.log_archives` setting, and change the location by specifying a new value for `basefilepath.log_archive`. For more information, see tsm File Paths.

4. After you generate the snapshot, you can select it and upload it to Technical Support, download it to your local machine, or delete it:
Tableau Server on Linux Administrator Guide

5. To download a local copy of the snapshot or to delete it, select the snapshot under Log Files and then select the appropriate Action.

Uploading log snapshots for Tableau Support

1. Click the Maintenance tab.

2. Select the snapshot you want to send.

3. Click Upload to Technical Support Case.

4. In the dialog that displays, enter the Support Case Number and your Contact
Email Address, then click **Upload Snapshot**.

Use the TSM CLI

You create a snapshot archive of Tableau Server log files using the `tsm maintenance ziplogs` command.

By default, this command creates a zip file containing all of the log files. If you are running a distributed installation of Tableau Server, perform this step from the initial node. Logs from all nodes will be included in the zip file.

**Note:** If you cannot run the ziplogs command successfully, you can manually zip the Tableau Server logs. For more information, see Troubleshoot Tableau Server on Linux.

To create a log file snapshot:
1. On the initial node, open a terminal session.

2. Type the following command:

   `tsm maintenance ziplogs -l -f <filename>`

   where `<filename>` is name of the zipped archive file you want to create. Choose a unique name with no spaces. If an existing ziplog with the same file name already exists, the creation of the file will fail unless you include the `-o` option to force an overwrite, delete the existing file, or specify a different name in the command.

   You can specify a time range for the snapshot and you can also specify which types of logs to include. For more information, see `tsm maintenance ziplogs`.

   The log file snapshot is saved to a fixed location on the computer where TSM and Tableau Server are installed. If you have a multi-node installation, the snapshot is saved to the initial node of the cluster. The location is specified by the `basefilepath_log_archives` variable.

   By default the log file snapshot is saved to:

   `/var/opt/tableau/tableau_server/data/tabsvc/files/log-archives`

   You can find the current location by querying the `basefilepath.log_archives` setting, and change the location by specifying a new value for `basefilepath.log_archive`. For more information, see `tsm File Paths`.

Sending log archives to Tableau Support

You can send log files to Tableau Support as part of a customer support case (a customer support case number is required). Before sending a log file, use `tsm maintenance ziplogs` command to combine the log files into a single zip file archive. If you are creating the archive to send to Tableau Support, see the Knowledge Base for information about how to upload large files.
In a terminal session, type the following command:

```
tsm maintenance send-logs -f <zip file name> -c <case number> -e <email address>
```

where `<case number>` is your support case number, `<email address>` is your contact email for this support case, and `<zip file name>` is the file name of your archive with .zip file extension.

### Change Logging Levels

By default, Tableau Server logs events at the **Info** level. You can change this if you need to gather more information (if you are working with Tableau Support, for example). As a best practice you should not increase logging levels except when troubleshooting an issue.

Increasing the log level to debug or trace increases the amount of information being logged and can have a significant impact to performance. You should only set a logging level to debug when investigating a specific issue. Reproduce the issue and then reset the logging level back to info.

### Logging Levels

The following logging levels are listed in order of increasing amount of information logged:

- off
- fatal
- error
- warn
- info (the default)
- debug
- trace

**Note:** Logging levels for new TSM processes are set at info and cannot be changed easily. If you are working with Tableau Technical Support and they determine that logs for these processes are needed, they can help you change the logging level.
temporarily. Processes impacted by this limitation include: Administration Agent, Administration Controller, Client File Service, Service Manager, Database Maintenance, and Backup/Restore.

Change Logging Levels

Set logging levels for Tableau Server using one of several tsm configuration set commands. The command you use depends on which component of Tableau Server you want to change the logging level for.

<table>
<thead>
<tr>
<th>Configuration key</th>
<th>Location of affected logs</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt;process&gt;.native_api.log.level</td>
<td>/vizqlserver/*.txt</td>
</tr>
<tr>
<td>Valid process names are backgrounder, vizportal, vizqlserver, dataserver, or tabadmincontroller</td>
<td></td>
</tr>
<tr>
<td>vizportal.log.level</td>
<td>/vizportal/*.log</td>
</tr>
<tr>
<td>vizqlserver.log.level</td>
<td>/vizqlserver/*.log</td>
</tr>
</tbody>
</table>

For more information, see tsm configuration set Options.

You need to stop Tableau Server before changing the logging levels, and restart it afterward. If you are running a distributed installation of Tableau Server, set logging levels from the initial node.

To change the logging level:
1. Stop Tableau Server by typing:

   tsm stop

2. Set the logging level to by typing

   tsm configuration set -k <config.key> -v <config_value>

   where <config.key> is <process>.native_api.log.level, viz-portal.log.level, or vizqlserver.log.level

   and <config_value> is a valid logging level.

   Examples:

   • tsm configuration set -k backgrounder.native_api.log.level -v debug
   • tsm configuration set -k vizqlserver.log.level -v warn
   • tsm configuration set -k vizportal.log.level -v debug

3. Apply pending changes by running the tsm pending-changes apply command.

4. Start Tableau Server by running the following command:

   tsm start

Reset Logging Levels

After you gather the information related to the issue you are investigating, reset the logging levels so there is no lingering performance impact.

Reset the logging level back to its default (info) using the appropriate command with a -d option.

Examples:
Troubleshoot Tableau Server Install and Upgrade

Follow the suggestions in this topic to resolve common issues with Tableau Server. For additional troubleshooting steps based on process status viewed on the Status page, see Troubleshoot Server Processes.

General Troubleshooting Steps

Many Tableau Server issues can be addressed with some basic steps:

1. Make sure there is enough disk space on each computer running Tableau Server. Limited disk space can cause a failure to install, a failure to upgrade, or problems running Tableau Server.

2. Restart Tableau Server. Issues related to processes not fully started can be resolved by restarting Tableau Server in a controlled way. To restart Tableau Server, use the tsm restart command. This will stop all the processes associated with Tableau Server and then restart them.

3. Reindex Tableau Server. Issues related to indexing can be resolved by reindexing Tableau Server. To reindex Tableau Server, use the tsm maintenance reindex-search command. For more information, see Reindexing Tableau Server Search & Browse below.

4. Restart the computer on which Tableau Server is running. Some issues, such as those related to data source connectivity, can be resolved by restarting the server computer.
Common Tableau Server Install Issues

Installation logs location

The install log, app-install.log, is located in /var/opt/tableau/tableau_server/logs.

The upgrade log, app-upgrade.log, is located in /var/opt/tableau/tableau_server/logs.

Unable to log into TSM or Tableau Server (sign in screen redisplay after entering credentials)

Using Internet Explorer or Edge, if you enter your credentials into the TSM or Tableau Server sign-in screen and the page redisplay without signing you in, verify that the hostname or domain in your URL does not include an underscore (_). If the hostname or domain of the Tableau Server computer includes an underscore (_) Internet Explorer or Edge browsers will not set a cookie, so the page will redisplay without signing you in. To work around this, use "localhost" or the IP address of the computer in the URL. For example: https://localhost:8850.

For more information, see the Tableau Knowledge Base.

Multiple install attempts fail

If you attempt to install Tableau Server and the install fails, any subsequent installation attempts are likely to fail unless you run the tableau-server-oblitrate script to clean Tableau off the computer.

A failed install attempt can leave the computer in a state that causes subsequent attempts to also fail with errors that don't seem directly related to a previous install attempt. One possible error is:

Enabling and starting all services
+ services=(appzookeeper* tabadmincontroller* tabsvc* licenseservice* fnplicenseservice* tabadminagent*)
Tableau Server on Linux Administrator Guide

clientfileservice*)
+ systemctl_user enable appzookeeper_0.service 'tabadmincontroller*' 'tabsvc*' 'licenseservice*' fnpllicenseservice_0.service 'tabadminagent*' 'clientfileservice*' 
++ id -ru a_tabadminpoc 
+ local unprivileged_uid=222954 
+ su -l a_tabadminpoc -c 'XDG_RUNTIME_DIR=/run/user/222954 systemctl --user enable appzookeeper_0.service tabadmincontroller* tabsvc* licenseservice* fnpllicenseservice_0.service tabadminagent* clientfileservice*' 
Failed to execute operation: No such file or directory

To fix this problem, run the tableau-server-oblitrate script to clean up any leftover remnants of the previous install attempt and then restart the computer. For more information, see Running the tableau-server-oblitrate script.

Important: If you created a backup of Tableau (<file>.tsbak) you want to keep (for example, to restore to your new installation), copy that file to a safe location on another computer to guarantee it is not removed when you clean up your Tableau computer.

Install fails due to hardware requirements

Tableau Server cannot install if the computer you are installing on does not meet the minimum hardware requirements. The requirements apply to all computers on which you are installing Tableau Server. For details on minimum hardware requirements, see Minimum Hardware Requirements and Recommendations for Tableau Server.

Common Tableau Server Upgrade Issues

Upgrade logs location
The upgrade log, app-upgrade.log, is located in /var/opt/tableau/tableau_server/logs.
Upgrade multi-node, initializing additional node fails with "Enter your credentials again" error

If you attempt to initialize an additional node when upgrading Tableau Server and see this error:

Enter your credentials again. The credentials you enter must provide administrative access to the computer where you generated the configuration file.

This is an indication that the node is unable to connect to or communicate with the initial node. This can happen for multiple reasons:

- The credentials you entered are not valid or you mistyped them. The credentials must be for a user who has administrative permissions on the computer where Tableau Server was first installed. You do not need to use the credentials of the user who created the bootstrap file but doing so will ensure you are using valid credentials.

- The local firewall of the computer you are trying to add is not allowing communication to the initial node. For more information, see Local firewall configuration.

- The nodes are on different subnets. In distributed installations, all nodes should be installed on the same subnet. For more information, see Distributed Requirements.

Upgrading fails due to lack of disk space

If there is not enough disk space for the Tableau Server Setup program to run and do the upgrade, the installation will fail. The amount of disk space required will depend on the size of your repository database and the number and size of your extracts. As a part of upgrading to version 9.0, the Setup program migrates extracts to the new File Store and this takes space.

To free up disk space:
1. Create a log archive snapshot using the `tsm maintenance ziplogs` command. After you create the ziplogs file, save it to a safe location that is not part of your Tableau Server installation.

2. Clean up unnecessary files using the `tsm maintenance cleanup` command. For more information, see Remove Unneeded Files.

Troubleshooting connections to TSM

Starting Tableau Server

Tableau Server cannot determine if it fully started

In some instances Tableau Server may report that it could not determine if all components started properly on startup. A message displays: "Unable to determine if all components of the service started properly."

If you see this message after starting, verify that Tableau Server is running as expected by using a `tsm status -v` command.

If the status shows as running ("Status: RUNNING"), then the server successfully started and you can ignore the message. If the status is DEGRADED or STOPPED, see "Tableau Server doesn't start" in the next section.

Tableau Server doesn't start

If Tableau Server does not start or is running in a degraded state, run the `tsm restart` command from a command prompt. This will shut down any processes that are running, and restart Tableau Server.

Reindexing Tableau Server Search & Browse

Problems that can be solved by rebuilding Search & Browse index

Symptoms of an index that needs to be rebuilt include:
A blank list of sites when a user attempts to log in
A blank list of projects when a user tries to select a project
Missing content (workbooks, views, dashboards)
Unexpected or inaccurate alerts (for example, an "refresh failed" alert on a workbook that does not include an extract)

If you see any of these behaviors, reset and rebuild the Search & Browse index using the `tsm maintenance reset-searchserver` command.

## Activating Tableau Server

**Tableau Server license activation fails**

In some instances Tableau Server license activation may fail. Error messages can range from a very generic one:

- **An error has occurred**

To more specific messages:

- **Function flxActCommonLicSpcPopulateFromTS returned error 50030, 71521,**

- **No license found for 'Tableau Server'**

To resolve this issue, try these solutions in the order listed:

**Confirm you can access the licensing server**

The Tableau licensing service was moved to a new data center on October 6, 2018. This means any environments that required special configuration (static IP safe listing for example) to access licensing.tableau.com or licensing.tableau.com will need to be updated before you can activate, refresh, or deactivate a Tableau product key.

**Tableau Server needs to connection to the following internet locations for licensing purposes:**
Tableau Server on Linux Administrator Guide

- licensing.tableau.com:443
- atr.licensing.tableau.com:443
- s.ss2.us
- ocsp.rootg2.amazontrust.com
- ocsp.rootca1.amazontrust.com
- ocsp.sca1b.amazontrust.com
- crt.sca1b.amazontrust.com
- crt.rootca1.amazontrust.com
- ocsp.sca0a.amazontrust.com
- crt.sca0a.amazontrust.com
- ocsp.sca1a.amazontrust.com
- crt.sca1a.amazontrust.com
- ocsp.sca2a.amazontrust.com
- crt.sca2a.amazontrust.com
- ocsp.sca3a.amazontrust.com
- crt.sca3a.amazontrust.com
- ocsp.sca4a.amazontrust.com
- crt.sca4a.amazontrust.com
- crl.rootca1.amazontrust.com
- crl.rootg2.amazontrust.com
- crl.sca1b.amazontrust.com

Requests to the above domains may be on port 80 or 443.

Verify the date and time

Verify the date and time on the initial Tableau Server computer is correct. If the clock is set to a time and date earlier than the current date, Tableau Server cannot be activated.

Force the product key to be read again

1. On the initial Tableau Server computer, sign in as administrator and open a command prompt.

2. Change to the Tableau Server bin directory. By default this is:

   ```
   /opt/tableau/tableau_server/packages/bin.<version_code>/
   ```

3. Type the following commands:

   ```
   tsm stop
   ./lmreread
   tsm start
   ```

Send the contents of trusted storage to Tableau Support

If FlexNet Licensing Services is installed and running but you’re still seeing an error, there might be a problem with the Tableau product key information. To resolve this issue, complete the following steps to create a file of the key information located in trusted storage.

1. On the initial Tableau Server computer, sign in as administrator and open a command prompt.

2. Type the following command:

   ```
   serveractutil -view > <machine_name>-LicResults.txt
   ```
This creates the `<machine_name>-LicResults.txt` file in your current directory. If you don’t have write permissions for that location and see an error, change to a location where you do have permission to create a file and run the command again.

3. Contact Tableau Support (http://www.tableau.com/support/request) and include the `<machine_name>-LicResults.txt` file that you created.

Troubleshoot Server Sign in Problems

There are several different sign in options between Tableau Services Manager (TSM) and Tableau Server.

- **TSM**—If you are not able to sign into TSM, make sure you are using credentials for a user who has administrative rights to the computer where TSM is installed. This user may or may not also be a Tableau Server administrator. This is true whether you are signing in to the Web UI or the CLI. For more information, see Sign in to Tableau Services Manager Web UI.

- **Tableau Server**—
  - Administrators: if you are signing into Tableau Server as an administrator, you must use credentials for a user who has an administrator role in Tableau Server. You create the initial administrator when you first install Tableau, but can add other users as administrators once Tableau is installed and running. For more information, see Sign in to the Tableau Server Admin Area.
  
  - Non-administrative users: If you are signing into Tableau Server as a user, you need to use credentials for a user who has been added to Tableau Server. For more information, see Sign in to Tableau Server or Online.

**Note:** If users with valid credentials are unable to sign into Tableau Server, make sure you have not added a node without applying pending changes.
If you have a pending new node, signing into Tableau Server may not be possible.

Troubleshooting scenarios

Unable to log into TSM or Tableau Server (sign in screen redisplays after entering credentials)

Using Internet Explorer or Edge, if you enter your credentials into the TSM or Tableau Server sign-in screen and the page redisplays without signing you in, verify that the hostname or domain in your URL does not include an underscore (\_). If the hostname or domain of the Tableau Server computer includes an underscore (\_) Internet Explorer or Edge browsers will not set a cookie, so the page will redisplay without signing you in. To work around this, use "localhost" or the IP address of the computer in the URL. For example: https://localhost:8850.

For more information, see the Tableau Knowledge Base.

Troubleshoot Licensing

This topic includes instructions for troubleshooting issues related to Tableau Server licensing.

Handle an unlicensed server

Tableau offers two licensing models: user-based and core-based. To learn more about user-based and core-based licensing, see Licensing Overview.

User-based licensing requires each active user account to be covered by a license. User-based licenses have a defined capacity, or number of users that they allow. Each user is assigned a unique user name on the server and is required to identify themselves when connecting to the server.
Core-based licensing has no constraints on the number of user accounts in the system, but it does restrict the maximum number of processor cores that Tableau Server can use. You can install Tableau Server on one or more machines to create a cluster, with the restriction that the total number of cores in all the machines does not exceed the number of cores you have licensed and that all of the cores on a particular machine are covered by the license.

Unlicensed User-Based Server

The most common reason for a server that has user-based licensing to be unlicensed is an expired product key or an expired maintenance contract.

Unlicensed Core-Based Server

A core-based server can become unlicensed for a variety of reasons, such as when Tableau Server nodes running licensed processes cannot contact the Tableau Server node running the License Manager service. To learn more about licensed processes, see Tableau Server Processes.

When the server is unlicensed you may not be able to start or administer the server. You can, however, manage your licenses using the tsm licenses command.

Troubleshoot user-based licensing

This section provides information about resolving issues that can occur when adding the user-based Viewer, Explorer and Creator licenses to Tableau Server or Tableau Online, or when these licenses expire. The highest available license type is Creator, followed by Explorer, and finally Viewer. To learn more about user-based licensing, see Licensing Overview.

A user or administrator is unlicensed due to license expiration

To avoid having users unexpectedly become unlicensed or move to another site role, you should always do one of the following before the license that they are currently using expires:

- Renew and activate a replacement license. If a user occupies a Creator, Explorer or viewer license and their license expires, they will use another license of the same type,
if available.
- Change the site role of those users to allow the use of a license that is not due to expire.

To learn how site roles can be changed to require a different license, see Set Users’ Site Roles.

The reassignment of users to new licenses is governed by the following logic:

- When a Server Administrator user occupies a Creator license and their license expires (with no replacement licenses available), they are reassigned to an Explorer license if any Explorer licenses are available. This license reassignment occurs in order of most recent login. Server Administrators displace other users who might be currently using an Explorer license. If no Creator or Explorer licenses are available a Server Administrator becomes unlicensed.

- When a non-Server Administrator user occupies a Creator license and their license expires (with no replacement licenses available), they become unlicensed. To avoid having these users become unlicensed, change their site role prior to license expiration. This is especially important for users in the Site Administrator Creator site role, who must move to the Site Administrator Explorer site role before their Creator license expires to avoid losing Site Administrator capabilities.

- When a non-Server Administrator user occupies an Explorer or Viewer license and their license expires (with no replacement licenses available), they are upgraded to a higher license type, if licenses of that type are available. Specifically, the following occurs when a license expires:
  - Users who occupy an Explorer license will move to a Creator license, if available (with no change to site role).
  - Users who occupy a Viewer license will move to an Explorer license, if available. If no Explorer licenses are available, these users will move to a Creator license, if available (with no change to site role).
  - If no licenses are available at the higher license types, those users are moved to Unlicensed.

Users are reassigned to a new license as described above in order of most recent login, with lower license types reassigned first (first Viewer, then Explorer, and then Creator).
Tableau Server on Linux Administrator Guide

For example: Two users with a Viewer license, a user with the Creator license, and two Server Administrators with a Creator license all have their licenses expire. Four unexpired Explorer licenses are available for these users. In this situation, the following occurs in the order shown below:

1. The user with a Viewer license who logged in most recently is reassigned to an Explorer license.
2. The second user with a Viewer license is reassigned to an Explorer license.
3. The Server Administrator user with a Creator license who logged in most recently is reassigned to an Explorer license, and then the second Server Administrator with a Creator license is reassigned to the remaining Explorer license.
4. The user with the Creator license becomes unlicensed.

Server Administrator site role is unchanged when using a Creator license

Server Administrators gain Creator capabilities if Creator licenses are available in Tableau Server, with no change to their site role name. All other Tableau Server and Tableau Online users gain Creator licenses only if assigned to a site role that includes Creator in its name.

Licenses are not immediately available

When you add a role-based license to Tableau Server, those licenses become available to all users when you restart Tableau Server. If you don't restart Tableau Server, role-based licenses become available to all users within 24 hours.

A user with a Viewer license cannot open Tableau Server or Tableau Online workbooks from Tableau Desktop

A user with a Viewer license who also has a separate Tableau Desktop license will be unable to open workbooks on Tableau Server or Tableau Online using Tableau Desktop. To open workbooks such using Tableau Desktop, that user will need an Explorer or Creator license on Tableau Server or Tableau Online.

Handle an Unlicensed VizQL Server Process

There are several status indicators on the Tableau Server Status page that help you understand the state of Tableau Server processes. An orange-color status box, "Unlicensed",
indicates that one of the VizQL server processes is unable to retrieve the Tableau Server license information.

There may be several reasons why the process is unable to access this information. For example, there may be network issues preventing a VizQL process, which is running on an additional node, from communicating with the initial node machine. Or, the process may be getting sent more requests than it can accept at that time and can’t handle the licensing request. As a result, some of your users may be able to access views while others cannot.

To resolve the problem, stop, then start Tableau Server.

**Tableau Services Manager (TSM) Command Timeout**

When Tableau Server is configured with two instances of the repository and failover to the backup repository occurs, TSM attempts to restart the original repository so that it is available as a backup. If this cannot be done for any reason, subsequent TSM commands can fail due to timeouts while waiting for the original repository to recover.

Commands that can be impacted include:
Tableau Server on Linux Administrator Guide

- tsm maintenance restore
- tsm maintenance reindex-search
- tsm reset
- tsm security regenerate-internal-tokens
- tsm sites export
- tsm sites import

If any of these commands is failing, and you have a repository that is not recovering, remove the repository from the server topology, apply pending changes, and re-add it.

Tableau Services Manager (TSM) Backup Timeout

When you back up Tableau Server, one of the first steps taken is to confirm that key services are running, and, if they are not, to start them. If these services cannot be started:

- Active Repository
- File Store
- Cluster Controller

any attempt to back up Tableau Server will fail with one of the following errors:

An error occurred starting one or more of the following services: Active Repository, File Store, Cluster Controller.

One or more of the following services did not start in a timely fashion: Active Repository, File Store, Cluster Controller.

To successfully back Tableau Server up, make sure these processes can start.

Cookie Restriction Error

When a user signs in to Tableau Server, a session cookie is stored in their local browser. The stored cookie is how Tableau Server maintains that the signed in user has been authenticated and can access the server. Because the cookie is set with the same domain or subdomain as the browser’s address bar, it is considered a first-party cookie. If a user’s browser is configured to block first-party cookies, they will be unable to sign in to Tableau Server.
When a user signs in to Tableau Server via an embedded view, or in an environment where trusted authentication has been configured, the same thing happens: a cookie is stored. In this case, however, the browser treats the cookie as a third-party cookie. This is because the cookie is set with a domain that’s different from the one shown in the browser’s address bar. If a user’s web browser is set to block third-party cookies, authentication to Tableau Server will fail. To prevent this from occurring, web browsers must be configured to allow third-party cookies.

Troubleshoot Subscriptions

"The view snapshot in this email could not be properly rendered."

If you receive a subscription with this error message, there could be several reasons:

- **Missing credentials**: Some views are published with embedded credentials. You may receive the above error if the embedded credentials are now out-of-date, or if the view was republished without the embedded credentials.

- **Database temporarily down**: If the view has a live database connection and the database was temporarily down when the subscription was being generated, you might receive the above error.

- **Background process timeout**: By default, the background process that handles subscriptions times out after 30 minutes. In the majority of cases, this is plenty of time. However, if the background process is handling an extraordinarily large and complex dashboard, that may not be enough time. You can check the Background Tasks for Non Extracts admin view to see if that’s the case. To increase the timeout threshold, use the `tsm configuration set subscriptions.timeout` command.

Can't see images in email

For images of content to display in a subscription email, users subscribed to views, in addition to View permissions, must also have Download Image/PDF permissions. For more information, see Permission capabilities.
Can't subscribe

If you can see a view on Tableau Server and it has a subscription icon (acağı) in the upper right corner, you can subscribe to it.

To subscribe to a view, Tableau Server needs to be correctly configured (described in Manage Subscriptions) and the view you’re subscribing to must either have embedded credentials for its data source or not rely on credentials at all. Examples of the latter include a workbook that connects to an extract that isn't being refreshed, or a workbook whose data is in a file that was included with the workbook at publish time. Embedding credentials is a step that happens in Tableau Desktop (see the Tableau Help for details).

No subscription icon

It's possible to see a view but be unable to subscribe to it. This can happen for several reasons:

- **No subscriptions have been scheduled**: If no subscriptions have been scheduled, the subscription icon will not appear. To set a schedule for subscriptions, see Create or Modify a Schedule.
- **The view uses a live database connection**: Views with live database connections, where you're prompted for your database credentials when you first click the view, aren't available for subscription. A subscription includes a view (or workbook), data, and a schedule. To deliver the data required for the view, Tableau Server either needs embedded database credentials or data that doesn't require credentials. Where live database connections are concerned, Tableau Server doesn't have the credentials, only the individual users do. This is why you can only subscribe to views that either don’t require credentials or have them embedded.
- **Tableau Server is configured for trusted authentication**: You may also be able to see a view but be unable to subscribe to it (no subscription icon) if Tableau Server is configured for trusted authentication. See Ensure Access to Subscriptions for more information.
Receiving invalid or "broken" subscriptions

If you configured subscriptions on test or development instances of Tableau Server in addition to your in-production instance, disable subscriptions on your non-production instances. Keeping subscriptions enabled on all instances can result in your users receiving subscriptions that appear to be valid, but which don't work, or receiving subscriptions even though they've unsubscribed from the view or workbook.

Missing PDF attachment

You can add a PDF attachment to your subscription if your administrator has it enabled. If the PDF attachment is missing from your subscription, it might because the size of the PDF exceeds either the email server size limit or the maximum size limit set by server administrators. In Tableau Server, the maximum size limit for PDF attachments to subscriptions can be adjusted through the tsm configuration set option subscriptions.max_attachment_size_megabytes. For more information, see Configure Server Event Notification and Set Up a Site for Subscriptions.

Suspended Subscriptions

By default, a subscription is suspended after 5 consecutive subscription failures. To change the threshold number of subscription failures that can occur before they are suspended, use the tsm configuration set option, backgrounder.subscription_failure_threshold_for_run_prevention. This sets the threshold for the number of consecutive failed subscriptions necessary before suspending the subscription. This is a server-wide setting.

Only Server administrators can configure the threshold number of subscription failures before a subscription is suspended. For information on setting this threshold, see Set up a Server for Subscriptions.

By default, administrators are not emailed when a subscription is suspended, but can opt-in to suspension emails per site through My Account Settings.
Resume and subscription owners can resume subscriptions in several ways:

- from My Subscription tab in Content Settings
- from the Subscriptions tab per workbook
- from the Subscriptions tab under Tasks (Server Admins only)

When a subscription is resumed, the alert failing count goes back to zero. The next evaluation of the subscription will occur at the next scheduled evaluation time.

Subscriptions not arriving ("Error sending email. Can't send command to SMTP host.")

You may see the above error in Windows Event Viewer if subscriptions aren't arriving and your SMTP server is using encrypted (SSL) sessions. Subscriptions are only supported for unencrypted SMTP connections. The solution is to use an unencrypted SMTP server. (If you're experiencing this error, note that Tableau Server will still indicate that subscriptions are being sent in the Background Tasks for Non Extracts admin view.)

Custom scripts not working after upgrade to 8.1

To support better session management, starting with version 8.1, a hash tag (#) was added to the end of view URLs. If you had custom subscriptions scripting that generated views as PDFs or PNGs you may need to update your scripts to allow for the hash tag.

For example, prior to version 8.1, view URLs use this syntax: `http://t-ableauserver/views/SuperStore/sheet1`. To generate a view as a PNG, `.png` could be added to the end of the URL. For example, `http://t-ableauserver/views/SuperStore/sheet1.png`.

In versions 8.1, 8.2, or 8.3, view URLs use this syntax: `http://t-ableauserver/views/SuperStore/sheet1#1`. To generate a PNG, add `.png`
before the hash tag. For example: http://t-
ableauserver/views/SuperStore/sheet1.png#1

Custom scripts not working after upgrade to 9.0

In version 9.0, the session ID at the end of server URLs is indicated by an "iid" parameter, :iid=<n>. For example, http://-localhost/#/views/Sales2015/SalesMarginsByAreaCode?:iid=1. This parameter replaces the hash tag "#<n>" used for the session ID in 8.x versions of Tableau Server.

If you use custom subscriptions scripts that generate views as PDFs or PNGs, you may need to update your scripts by removing the hash tag and number (#<n>), and inserting the ?:iid= session ID parameter before the number.

Starting in version 9.0, view URLs use this syntax: http://t-
ableauserver/views/SuperStore/sheet1?:iid=2.

To generate a PNG in version 9.0 and later, add .png before the session ID: http://t-
ableauserver/views/SuperStore/sheet1.png?:iid=2

Troubleshoot Performance of Workbooks with Calculations

In most cases, you can expect to see similar or better performance after upgrading your Tableau Server from versions 10.4 or earlier to 10.5 or later. However, for workbooks that use complex calculations, you may see performance issues after you upgrade. This article describes how to test and troubleshoot performance issues for workbooks that have calculations and use extracts as the data source after you have upgraded your Tableau Server from versions 10.4 or earlier to 10.5 or later.

If you have not yet upgraded to Tableau Server 10.5 or later, we strongly recommend that you set up a test environment and test your workbook performance before you upgrade your production Tableau Server. For more details, see Test Workbook Performance.
If you have already upgraded to Tableau Server 10.5 or later, and you are experiencing slow response times, use the guidance provided in this topic to help troubleshoot performance issues.

Since Linux was first released in 10.5, this only applies when you are migrating from Tableau Server versions 10.4 or earlier on Windows to Linux 10.5 or later.

Here are the scenarios that most likely describe your environment after upgrading to Tableau Server 10.5 or later:

- Recently upgraded, and majority of the extracts are still in .tde format
- Recently upgraded and majority of extracts are in .hyper format
- Upgraded a while ago, workbooks have changed since upgrade

Scenario: You upgraded your Tableau Server recently. You are experiencing slow response times and the majority of your extracts have not been refreshed, so they are still in .tde format.

1. Disable any scheduled or automated extract refreshes.

2. Enable the following setting on Tableau Server to check the number of query nodes required for Tableau to generate the workbook and display an error when the set limit is exceeded. The error message might say something like: "Logical Query tree has 2348182 nodes, maximum number allowed is 1000000....".

Use TSM CLI to run the following commands:

```
tsm configuration set -k native_api.node_limit_checker_pre_rewrite_disable -v false
```
```
tsm pending-changes apply
```

**Note:** This setting is only available in Tableau Server 10.5.3 or later. While enabling this setting improves the reliability and stability of Tableau Server, you
may see errors on a very small percentage of workbooks when you enable this setting. For example, the Tableau team saw errors on 0.01% of the workbooks with this setting enabled on Tableau Online.

3. Test the workbooks by viewing or interacting with them. You should see an error message informing you about exceeding the node limits on workbooks that have performance issues. The error message might say something like: "Logical Query tree has 2348182 nodes, maximum number allowed is 1000000....".

4. Use Tableau Desktop to troubleshoot your workbook performance. Run this command to enable the node limit check in your Tableau Desktop:

```
tableau.exe -DNodeLimitCheckerPreRewriteDisable=false
```

**Note:** This setting is only available in Tableau Desktop 10.5.3 or later.

In Tableau Desktop, download and open the workbooks that displayed the error message about high node limits in step 2. Try optimizing the performance of workbooks using methods described in *Best Practices for Creating Calculations in Tableau*. If successful, publish these workbooks to Tableau Server. If you are still experiencing performance issues, continue to the next step.

5. For workbooks that load successfully, but the overall response times is slower after upgrade, try using the Compute Calculations Now option. If you have a large number of workbooks that fall under this category, consider enabling the server-wide setting to retain materialized calculations for all extracts when they are upgraded from .tde to .hyper:

Use TSM CLI to run the following commands:
Considerations for turning on this setting:

Turn this setting on only if the majority of your workbooks require calculations to be materialized. There are resource and time costs associated with this setting, which are as follows:

- When this setting is set to true, the extract file size will increase, affecting the overall disk space.
- Extracts with materialized calculations take a longer time to refresh than extracts without materialized calculations.

This setting only affects extracts that have not yet upgraded to .hyper. Extracts already upgraded to .hyper before turning on this setting, will not have the calculations materialized. This setting will also not affect workbooks and extracts published from Tableau Desktop, or any new extract refreshes published to Tableau Server that do not use the Compute Calculations Now option.

**Note:** This setting is only available in Tableau Server 10.5.3 or later.

6. Enable extract refresh schedules.
Scenario: You upgraded your Tableau Server recently. You are experiencing slow response times and a majority of your extracts have been refreshed and upgraded from .tde to .hyper.

1. Restore the backup you made before you upgraded your production Tableau Server.

   **Note:** This method assumes that your workbooks have not changed significantly since the upgrade and, or it is OK to go back to the state just before upgrade.

2. Disable any scheduled extract refreshes.

3. Enable the following setting on Tableau Server to check the number of query nodes required for Tableau to generate the workbook and display an error when the set limit is exceeded. The error message might say something like: "**Logical Query tree has 2348182 nodes, maximum number allowed is 1000000.**"

   Use TSM CLI to run the following commands:

   ```
   tsm configuration set -k native_api.node_limit_checker_pre_rewrite_disable -v false
   tsm pending-changes apply
   ```

   **Note:** This setting is only available in 10.5.3 or later. While enabling this setting improves the reliability and stability of Tableau Server, you may see errors on a very small percentage of workbooks with when you enable this setting. For example, the Tableau team saw errors on 0.01% of the workbooks with this setting enabled on Tableau Online.

4. Test the workbooks by viewing or interacting with them. You should see an error message informing you that you are exceeding the node limits on workbooks that have
performance issues. The error message might say something like: "**Logical Query tree has 2348182 nodes, maximum number allowed is 1000000...**."

5. Use Tableau Desktop to troubleshoot your workbook performance. Run this command to enable the node limit check in your Tableau Desktop:

```bash
tableau.exe -DNodeLimitCheckerPreRewriteDisable=false
```

Download and open the workbooks that displayed the error message using Tableau Desktop. Try optimizing the performance of workbooks using methods described in *Best Practices for Creating Calculations in Tableau*. If successful, then publish them to Tableau Server. If you are still experiencing performance issues, continue to the next step.

6. For workbooks that load successfully, but the overall response time is slower after upgrade, try using the **Compute Calculations Now** option. If you have a large number of workbooks that fall under this category, consider enabling the server-wide setting to retain materialized calculations for all extracts when they are upgraded from .tde to .hyper:

Use TSM CLI to run the following commands:

```bash
tsm configuration set -k native_api.preserve_calculations_on_hyper_refresh_conversion -v true
tsm pending-changes apply
```

**Considerations for turning on this setting:**

Turn this setting on only if majority of your workbooks require calculations to be materialized. There are resource and time costs associated with this setting, which are as follows:
• When this setting is set to true, the extract file size will increase, affecting the overall disk space.

• Extracts with materialized calculations take a longer time to refresh than extracts without materialized calculations.

This setting only affects extracts that have not yet upgraded to .hyper. Extracts already upgraded to .hyper before turning on this setting will not have the calculations materialized. This setting will also not affect workbooks and extracts published from Tableau Desktop, or any new extract refreshes published to Tableau Server that do not use the Compute Calculations Now option.

**Note:** This setting is only available in 10.5.3 or later.

7. Enable extract refresh schedules.

Scenario: You upgraded your Tableau Server a while ago and you are experiencing slow response times. It is not practical for you to roll back using the backup you made prior to upgrade because your workbooks have changed since the upgrade, and the extracts have been upgraded to .hyper format.

If you fall under this scenario, and you find that your workbooks require materialized calculations, turning on the optimization setting on refresh will not work for you as this setting only works when upgrading from .tde to .hyper. You will need to test your workbooks individually for performance issues, and optimize them manually and republish to Tableau Server.

1. If you see high memory and CPU consumption by VizQL Server when viewing workbooks with extracts, in addition to slower workbook response times, then you can enable the following setting on Tableau Server to help prevent the VizQL process from consuming the memory resources on the machine that might result in critical
server issues. This setting checks the number of query nodes required for Tableau to generate the workbook and displays an error when the set limit is exceeded:

Use TSM CLI to run the following commands:

```
tsm configuration set -k native_api.node_limit_checker_pre_rewrite_disable -v false

 tsm pending-changes apply
```

**Note:** While enabling this setting improves the reliability and stability of Tableau Server, you may see errors on a very small percentage of workbooks when you enable this setting. For example, the Tableau team saw errors on 0.01% of the workbooks with this setting enabled on Tableau Online.

2. Test the workbooks by viewing or interacting with them. Look for workbooks where you might see an error message that says something like: "**Logical Query tree has 2348182 nodes, maximum number allowed is 1000000....**".

3. Use Tableau Desktop to troubleshoot the workbooks that you identified in step 2. Run this command to enable the node limit check in your Tableau Desktop:

```
tableau.exe -DNodeLimitCheckerPreRewriteDisable=false
```

Download and open the workbooks that displayed the error message using Tableau Desktop. Try optimizing the performance of workbooks using methods described in the Best Practices for Creating Calculations in Tableau topic and publish them to Tableau Server. If you are still seeing performance issues, continue to the next step.

4. Use the **Compute Now** option to materialize calculations in your extract.
Server Administrator Reference

You can learn more about Tableau Server processes, ports, and accounts and permissions.

Tableau Server Processes

This topic describes the options for setting the process configuration. To configure Tableau Server processes, you need to specify which processes and how many instances should run on each node. You do this using the tsm topology set-process command. For more information, see Changing the number of processes on a node.


Except where explicitly noted in the table below, applying changes in processes will stop Tableau Server if it is running when you apply those changes. After changes are applied, Tableau Server is returned to the state it was in before process configuration, so if the server was running, it will be restarted.

Important: Your process topology will depend on your organizational needs.

Licensed processes

Some of the processes that are installed as a part of Tableau Server are "licensed" processes. Licensed processes need a valid Tableau Server license in order to run. Other processes that are installed as a part of Tableau Server are not tied to a valid license. This has the following impact:

- Every licensed process needs to regularly contact the Tableau Server License Manager service that runs on the initial Tableau Server computer to verify they are
Tableau Server on Linux Administrator Guide

licensed. If they cannot confirm there is a valid license, for example, if the initial node is not available, the process will not run and Tableau Server may not function properly or reliably.

- If you have a core-based Tableau Server license, the cores on any node with a licensed process will count against the total count of licensed cores.

**Note:** If you have the Data Management Add-on and a core-based license, you will need to understand how the licensed processes will count against the total count of licensed cores that come with each license. For more information, see Licensing Tableau Prep Conductor for Tableau Server.

The "Licensed" column in the table below identifies those processes that require a valid license, and which impact the count of cores in core-based licenses.

<table>
<thead>
<tr>
<th>Name shown in tsm status -v</th>
<th>Name used with tsm topology set-process</th>
<th>Purpose</th>
<th>Notes</th>
<th>Lentiseded</th>
</tr>
</thead>
<tbody>
<tr>
<td>Application Server</td>
<td>vizportal</td>
<td>The Application Server (VizPortal) handles the web application, REST API calls, and supports browsing and searching.</td>
<td>When Application Server is installed, Data Engine is also installed, unless the node already has an instance of Data Engine.</td>
<td>Yes</td>
</tr>
</tbody>
</table>
### Ask Data

| Ask Data | Cannot be configured manually. | The Ask Data service is used by the Ask Data feature. | Runs automatically on all nodes where Data Server is running. | No |

### Backgrounder

| Backgrounder | backgrounder | The Backgrounder runs server tasks, including extract refreshes, subscriptions, ‘Run Now’ tasks, and tasks initiated from tabcmd. | When Backgrounder is installed, Data Engine is also installed, unless the node already has an instance of Data Engine. 

Backgrounder is a single-threaded process. You can add more instances of backgrounder to a node to expand the capacity of the node to run jobs in parallel.

In most situations you can change the number of backgrounder instances on an existing node of a running server without causing a stop and restart the server. An exception is if you are adding backgrounder to an existing | Yes |
| Cache Server | cachesserver | The Cache Server is a query cache distributed and shared across the server cluster. This in-memory cache speeds user experience across many scenarios. VizQL server, backgrounder, and data server (and application server to a lesser extent) make cache requests to | No |

For more information, see Tableau Server Dynamic Topology Changes.
<table>
<thead>
<tr>
<th>Component</th>
<th>Role</th>
<th>Description</th>
<th>Required on each node. Not automatically installed.</th>
<th>Yes/No</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cache Server</td>
<td>the cache server on behalf of users or jobs. The cache is single-threaded, so if you need better performance you should run additional instances of cache server.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cluster Controller</td>
<td>cluster-controller</td>
<td>The Cluster Controller is responsible for monitoring various components, detecting failures, and running failover when needed.</td>
<td>Required on each node. Not automatically installed.</td>
<td></td>
</tr>
<tr>
<td>Data Engine</td>
<td>Cannot be configured manually.</td>
<td>The Data Engine creates data extracts and processes queries.</td>
<td>Automatically installed when you install File Store, VizQL Server, Application Server (VizPortal), Data Server, or Back grounder.</td>
<td></td>
</tr>
<tr>
<td>Data Server</td>
<td>dataserver</td>
<td>The Data When Data Server is</td>
<td>Yes</td>
<td></td>
</tr>
<tr>
<td>Component</td>
<td>Description</td>
<td>Requirement</td>
<td>Note</td>
<td></td>
</tr>
<tr>
<td>-----------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>------------------------------------------------------------------------------</td>
<td>------</td>
<td></td>
</tr>
<tr>
<td>Elastic Server</td>
<td>Elastic Server is used by Ask Data to index data.</td>
<td>Elastic Server processes can be running on more than one node in a cluster. Optionally, they can be moved to any node. It is recommended to have an odd number of Elastic Server processes running. The Elastic Server heap size can be configured by using the <code>elasticsearch.vmopts</code> TSM configuration option. For more information, see <code>tsm configuration set Options</code>.</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>File Store</td>
<td>The File Store automatically replicates extracts across Data Engine</td>
<td>When File Store is installed, Data Engine is also installed, unless the node already has an instance of Data Engine.</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>Service</td>
<td>Description</td>
<td>Requirements</td>
<td>Notes</td>
<td></td>
</tr>
<tr>
<td>-------------------------</td>
<td>------------------------------------------------------------------------------</td>
<td>------------------------------------------------------------------------------</td>
<td>-------</td>
<td></td>
</tr>
<tr>
<td>Gateway</td>
<td>The Gateway is a web server that handles all requests to Tableau Server from browsers, Tableau Desktop, and other clients.</td>
<td>Required on any node with an instance of VizQL Server or Viz-portal.</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>Messaging Service</td>
<td>The Messaging Service is used to support communication between microservices in Tableau Server.</td>
<td>Automatically installed on initial node when you install Tableau Server. One instance of the service is required. On multi-node installations of Tableau Server, you can move the Messaging Service to a different node. You cannot configure more than one instance in a cluster. For more information, see Tableau Server Messaging Service.</td>
<td>No</td>
<td></td>
</tr>
<tr>
<td>Repository</td>
<td>The PostgreSQL repository is the</td>
<td>You are limited to a maximum of two instances of the repository.</td>
<td>No</td>
<td></td>
</tr>
</tbody>
</table>
### Tableau Server on Linux Administrator Guide

<table>
<thead>
<tr>
<th>Service</th>
<th>Description</th>
<th>Requirements</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>SAML Service</td>
<td>Cannot be configured manually.</td>
<td>The SAML Service acts as a proxy between Tableau Server and SAML Identity Providers (IdPs).</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Automatically installed on each node where you install Tableau Server.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Shows a status of stopped in output of tsm status -v unless site SAML is enabled.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>You cannot configure the SAML Service manually.</td>
<td></td>
</tr>
<tr>
<td>Search And Browse</td>
<td>searchserver</td>
<td>The Search Service handles fast search, filter, retrieval, and display of content metadata on the server.</td>
<td>No</td>
</tr>
<tr>
<td>Tableau Prep Con-</td>
<td>flowprocessor</td>
<td>The Tableau</td>
<td>Yes</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Requires Data Management.</td>
<td></td>
</tr>
<tr>
<td>Component</td>
<td>Version</td>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>--------------------</td>
<td>---------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td></td>
</tr>
<tr>
<td>Prep Conductor</td>
<td></td>
<td>Prep Conductor runs a flow, checks connection credentials, and sends alerts if a flow fails. It leverages the scheduling and tracking functionality of Tableau Server so you can automate running flows to update the flow output.</td>
<td></td>
</tr>
<tr>
<td>VizQL Server</td>
<td>vizqlserver</td>
<td>The VizQL Server loads and renders views, computes and executes queries.</td>
<td></td>
</tr>
</tbody>
</table>

When VizQL Server is installed, Data Engine is also installed, unless the node already has an instance of Data Engine.

In most situations you can change the number of VizQL instances on an existing node of a...
running server without causing a stop and restart the server. An exception is if you are adding VizQL to an existing node that did not previously have VizQL or any other process that also installs Data Engine. For more information, see Tableau Server Dynamic Topology Changes.

**Tableau Microservice Container Processes** These processes are automatically added when one of the microservices they contain is added to a node. Container status depends on the status of the microservices within the container. If all microservices are running, the container process has a status of **running**. If all microservices are stopped, the container process status is **error**. If one or more microservices is running while others are not, the container service has a status of **degraded**. For more information, see Tableau Server Microservice Containers.

<table>
<thead>
<tr>
<th>Interactive Microservice Container</th>
<th>Container process for internal Tableau Server microservices that are bundled together for ease of</th>
<th>These containers and the microservices they contain cannot be manually configured. The microservices may change over time.</th>
<th>No</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Non-Interactive Microservice Container</strong></td>
<td>Container process for internal Tableau Server microservices that are bundled together for ease of deployment and scalability purposes.</td>
<td>These containers and the microservices they contain cannot be manually configured. The microservices may change over time.</td>
<td>No</td>
</tr>
</tbody>
</table>

**Tableau Services Manager (TSM) Processes**  These processes have a status of **running** once TSM has been initialized, and remain running even when Tableau Server is stopped.

| **Administration Agent** | Cannot be configured manually. | The TSM Agent monitors the Coordination Service for changes to configuration or topology and delivers new configurations to each service (configuration- or deploys | Automatically installed on each node where you install Tableau Server. You cannot configure the Administration Agent manually. | No |
### Administration Controller

<table>
<thead>
<tr>
<th>new services and removes old ones (topology)</th>
</tr>
</thead>
</table>

**Cannot be configured manually, except to move it to another node. For more information, see Recover from an Initial Node Failure.**

The TSM Controller handles requests to TSM and orchestrates configuration and topology changes and workflow across service processes. The Controller also serves as the REST API endpoint (HTTPS).

Automatically installed when you install TSM on the initial node.

You cannot configure the Administration Controller manually except to move it to another node. For more information, see Recover from an Initial Node Failure.

### Client File Service

<table>
<thead>
<tr>
<th>cli-entfileservice</th>
</tr>
</thead>
</table>

The Client File Service (CFS) manages most shared files in a multinode cluster. For example, authentication related cer-

Automatically installed on the initial node. No other instances are installed unless you explicitly configure them. See Configure Client File Service.

In multi-node deployments, we recommend you configure an
<table>
<thead>
<tr>
<th>Component</th>
<th>Setting/Configuration Required</th>
<th>Status/Impact</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Coordination Service</strong></td>
<td>Cannot be set with <code>tsm topology set-process</code>.</td>
<td>The Coordination Service serves as the single source of truth.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Automatically installed on the initial node. No other instances are installed unless you explicitly deploy a new Coordination Service ensemble. For details, see Deploy a Coordination Service Ensemble.</td>
</tr>
<tr>
<td><strong>Service Manager</strong></td>
<td>Cannot be configured manually.</td>
<td>The Service Manager</td>
</tr>
</tbody>
</table>
### Tableau Server on Linux Administrator Guide

| License Manager | Cannot be configured manually. | The License Manager handles licensing. | Automatically installed on the initial node when you install TSM. A single instance of this is installed on a Tableau Server cluster. The License Manager process should only be manually configured if the initial node fails. For more information, see Recover from an Initial Node Failure | No |

| Database Maintenance | Cannot be configured manually. | The Database Maintenance service is responsible for performing maintenance operations on the Tableau Server repository. | Automatically installed on each node where you install Tableau Server. Shows a status of **stopped** in output of `tsm status -v` unless it is actively performing database maintenance. Maintenance can include updates | No |

**Tableau Server Maintenance Processes** These processes have a status of **stopped** unless they are actively running to complete a job.
<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
<th>Status</th>
<th>Required?</th>
</tr>
</thead>
<tbody>
<tr>
<td>Backup/Restore</td>
<td>Cannot be configured manually.</td>
<td>The Backup and Restore service is responsible for performing backup and restore operations on the data stored in the Tableau Server repository and file store.</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Automatically installed on each node where you install Tableau Server.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Shows a status of <code>stopped</code> in output of <code>tsm status -v</code> unless it is performing a backup or restore operation.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>You cannot configure the Backup and Restore service manually.</td>
<td></td>
</tr>
<tr>
<td>Site Import/Export</td>
<td>Cannot be configured manually.</td>
<td>The Site Import and Export service is responsible for migrating</td>
<td>No</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Automatically installed on each node where you install Tableau Server.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Shows a status of</td>
<td></td>
</tr>
</tbody>
</table>
Tableau Server sites between server clusters.

stopped in output of tsm status -v unless it is performing an import or export.

You cannot configure the Site Import and Export service manually.

Process workflow

The Tableau Server processes and how they interact depend on what action or activity is taking place. For example, the processes that are used, and how they interact, differ when you publish a workbook from those used when you sign in using SAML. For some interactive views on process workflow, see the below workbook. This allows you to select a particular workflow and follow it step by step from start to finish.

Disclaimer: This workbook is published on Tableau Public and is not maintained by Tableau documentation. We cannot guarantee that it is up-to-date with the latest version of Tableau Server.
Tableau Server Administration Agent

The Administration Agent monitors the Coordination Service for changes to configuration or topology and delivers new configurations to each service (configuration) or deploys new services and removes old ones (topology). The Administration Agent also checks each of the services for status and reports this back to the Coordination Service. This process will be automatically configured for you for each node of the cluster during installation—no explicit configuration is required.

The Administration Agent may also be referred to as the TSM Administration Agent.

+-----------------+----------------------------------+
| **Process**     | Administration Agent             |
| **Status**      | Status of the Administration Agent process is not visible on the Status Page. |
Use the TSM CLI to view status. For more information, see View Server Process Status

| Logging | Logs generated by the Administration Agent process are located in /var/opt/tableau/tableau_server/data/tabsvc/logs/tabadminagent. For more information, see Server Log File Locations |

What happens when an Administration Agent process fails? All other Tableau Server processes running on the same node will display as “unavailable” on the TSM status page. Tableau Server will continue to work as expected, however you will not be able to make configuration/topology changes to the cluster. Failed Administration Agent processes automatically restart as long as the computer itself is otherwise healthy. If the Administration Agent doesn’t start up on the node, you can try to start the service manually by running the following command:

```
sudo su -l tableau

systemctl --user start tabadminagent_0
```

**Tableau Server Application Server**

The Application Server (VizPortal) handles the web application and REST API calls. Application Server also supports browsing and searching. To ensure high availability of Application Server, configure instances on each node in the Tableau Server cluster.

<table>
<thead>
<tr>
<th>Process</th>
<th>Application Server</th>
</tr>
</thead>
<tbody>
<tr>
<td>Status</td>
<td>Status of the Application Server process is visible on the Status Page. For more information, see View Server Process Status</td>
</tr>
<tr>
<td>Logging</td>
<td>Logs generated by the Application Server process are located in /var/opt/tableau/tableau_server/data/tabsvc/logs/vizportal. For more information, see Server Log File Locations</td>
</tr>
</tbody>
</table>
What happens when an Application Server process fails? Requests being handled by that instance will fail, but subsequent requests will be routed to other running Application Server processes. Assuming the node containing the failed Application Server is still running, the failed process should automatically restart within seconds.

### Tableau Server Backgrounder Process

The Backgrounder process runs server tasks, including extract refreshes, subscriptions, ‘Run Now’ tasks, and tasks initiated from tabcmd. To make the Backgrounder process highly available, you should configure one or more instances to run on multiple nodes in the cluster.

Backgrounder is single-threaded. By adding more Backgrounder instances to a node, you can increase the number of jobs that can be run in parallel on that node. You can add Backgrounder instances up to one half the number of cores. When deciding where and how many Backgrounders to run, consider how other server processes will affect each machine’s available capacity.

<table>
<thead>
<tr>
<th>Process</th>
<th>Backgrounder</th>
</tr>
</thead>
<tbody>
<tr>
<td>Status</td>
<td>Status of the Backgrounder process is visible on the Status Page. For more information, see View Server Process Status</td>
</tr>
<tr>
<td>Logging</td>
<td>Logs generated by the Backgrounder process are located in /var/‐opt/tableau/tableau_server/data/tabsvc/logs/back‐ grounder. For more information, see Server Log File Locations</td>
</tr>
</tbody>
</table>

What happens if a Backgrounder process goes down? Refresh and subscription jobs on the failed Backgrounder process are retried once the Backgrounder process recovers from failure. Most background jobs are scheduled to run periodically, and the same background task will be picked up and performed normally at the next scheduled time by a functioning Backgrounder process.

Failed Backgrounder processes automatically restart as long as the computer itself is otherwise healthy, and the failed jobs will be retried.
Tableau Server on Linux Administrator Guide

Related content

- Improving group synchronization performance

Tableau Server Cache Server

The Cache Server provides a shared external query cache. It’s a cache of key/value pairs which hold information from previous queries to speed up future requests. To make Cache Server highly available, configure one or more Cache Server processes on multiple nodes of the cluster.

<table>
<thead>
<tr>
<th>Process</th>
<th>Cache Server</th>
</tr>
</thead>
<tbody>
<tr>
<td>Status</td>
<td>Status of the Cache Server process is visible on the Status Page. For more information, see View Server Process Status</td>
</tr>
<tr>
<td>Logging</td>
<td>Logs generated by the Cache Server process are located in /var/-opt/tableau/tableau_server/data/tabsvc/logs/cacheserver. For more information, see Server Log File Locations</td>
</tr>
</tbody>
</table>

What happens when Cache Server process goes down? The consequences are relatively mild. Tableau Server will still work but actions may take longer as they do not have cached results available. As the queries rerun, the restarted Cache Server gets repopulated, eventually speeding things up for end users. In effect, Cache Server does not have an availability impact; however, it does have an impact on various end user performance scenarios. To reduce impact on user performance, run multiple processes of this type across the cluster.

A failed Cache Server process is automatically restarted; as long as the computer itself is otherwise healthy, the Cache Server process will relaunch.

Tableau Server Client File Service

The Client File Service (CFS) stores and distributes files needed by TSM (e.g. certificates, customization files, etc.). The following files are managed by CFS:
The following files are not managed or distributed by CFS:

- External SSL files. The certificate and key files for external SSL are stored and managed by the Coordination Service. You do not need to manually distribute these files.
- SSL files for LDAP identity store. You must distribute the SSL certificate file manually to each node in the cluster. See LDAP over SSL.

The Client File Service functions much like the File Store does for files needed by business services. By default, CFS is only installed on the initial node of your Tableau Server installation. To configure CFS to for high availability, we recommend that you configure an instance of CFS on each of the nodes where you deploy the Coordination Service.

<table>
<thead>
<tr>
<th>Process</th>
<th>Client File Service</th>
</tr>
</thead>
<tbody>
<tr>
<td>Status</td>
<td>Status of the Client File Service process is not visible on the Status Page. Use the TSM CLI to view status. For more information, see View Server Process Status</td>
</tr>
<tr>
<td>Logging</td>
<td>Logs generated by the Client File Service process are located in /var/opt/tableau/tableau_server/data/tabsvc/logs/cli-entfileservice. For more information, see Server Log File Locations</td>
</tr>
</tbody>
</table>
What happens when a CFS process fails? Nothing, as long as there is still at least one functioning CFS process in the cluster. The controller will redirect file transfer requests to the other working CFS process.

Failed Client File Service processes automatically restart as long as the computer itself is otherwise healthy.

Tableau Server Coordination Service

The Coordination Service is built on Apache ZooKeeper, an open-source project, and coordinates activities on the server, guaranteeing a quorum in the event of a failure, and serving as the source of “truth” regarding the server topology, configuration, and state. The service is installed automatically on the initial Tableau Server node, but no additional instances are installed as you add additional nodes. Because the successful functioning of Tableau Server depends on a properly functioning Coordination Service, we recommend that for server installations of three or more nodes, you add additional instances of the Coordination Service by deploying a new Coordination Service ensemble. This provides redundancy and improved availability in the event that one instance of the Coordination Service has problems.

The hardware for your cluster can have some effect on how well the Coordination Service runs. In particular:

- Memory. The Coordination Service maintains state information in memory. By design, the memory footprint is small, and is typically not a factor in overall server performance.

- Disk speed. Because the service stores state information on disk, it benefits from fast disk speed on the individual node computers.

- Connection speed between nodes. The service communicates continuously between cluster nodes; a fast connection speeds between nodes helps with efficient synchronization.
Status

Status of the Coordination Service process is not visible on the Status Page. Use the TSM CLI to view status. For more information, see View Server Process Status

Logging

Logs generated by the Coordination Service process are located in /var/opt/tableau/tableau_server/data/tabsvc/logs/app-zookeeper. For more information, see Server Log File Locations

Configuration for the Coordination Service

The Coordination Service is installed automatically on the initial node of Tableau Server. If you are running a single-node installation, you do not need to do anything to deploy or configure the Coordination Service. If your installation includes three or more nodes, you'll be prompted to configure a Coordination Service ensemble when you add your third node. This is not required, but is highly recommended as the Coordination Service serves a key function for high availability, acting as the source of "truth" about server topology, configuration, and state.

To configure a Coordination Service ensemble, use the TSM CLI and add the Coordination Service to the nodes you want running it. For details on how to deploy a Coordination Service ensemble, see Deploy a Coordination Service Ensemble.

The Coordination Service Quorum

To ensure that the Coordination Service can work properly, the service requires a quorum—a minimum number of instances of the service. This means that the number of nodes in your installation impacts how many instances of the Coordination Service you want to configure in your ensemble.

Number of Coordination Service instances to use

The maximum number of Coordination Service instances you can have in an ensemble on Tableau Server depends on how many Tableau Server nodes you have in your deployment. Configure a Coordination Service ensemble based on these guidelines:
<table>
<thead>
<tr>
<th>Total number of server nodes</th>
<th>Recommended number of Coordination Service nodes in ensemble (must be 1, 3, or 5)</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>1-2 nodes</td>
<td>1 node</td>
<td>This is the default and requires no changes unless you want to move the Coordination Service off your initial node and onto your additional node.</td>
</tr>
<tr>
<td>3-4 nodes</td>
<td>3 nodes</td>
<td>Five is the maximum number of Coordination Service instances you can install.</td>
</tr>
<tr>
<td>5 or more nodes</td>
<td>5 nodes</td>
<td></td>
</tr>
</tbody>
</table>

If you reduce the number of nodes

If you reduce the nodes in your cluster from three (or more) to two nodes, a warning tells you Tableau Server can no longer support high availability:

A minimum of three Tableau Server nodes are required for high availability. You can add a third node now, or continue with only two nodes. Continuing with only two nodes means Tableau Server will not be highly available. You can always add a third node later. Click OK to continue with 2 nodes, or Cancel to go back and add a node.

If you continue, Tableau Server will run, but you will not have any automatic failover of the repository.

Viewing Coordination Service Status

The Coordination Service is not included in the listing when you View Server Process Status. To see the state of the service, you can use the tsm status command:
tsm status -v

The output from the command shows you whether the service is running:

node1: TABLEAUSVR01
Status: RUNNING
'Tableau Server Gateway 0' is running.
'Tableau Server Application Server 0' is running.
'Tableau Server VizQL Server 0' is running.
'Tableau Server VizQL Server 1' is running.
'Tableau Server VizQL Server 2' is running.
'Tableau Server VizQL Server 3' is running.
'Tableau Server Cache Server 0' is running.
'Tableau Server Cache Server 1' is running.
'Tableau Server Coordination Service 0' is running.
'Tableau Server Cluster Controller 0' is running.
'Tableau Server Search And Browse 0' is running.
'Tableau Server Backgrounder 0' is running.
'Tableau Server Backgrounder 1' is running.
'Tableau Server Data Server 0' is running.
'Tableau Server Data Server 1' is running.
'Tableau Server Data Engine 0' is running.
'Tableau Server File Store 0' is running.
'Tableau Server Repository 0' is running (Active Repository).
'Tableau Server Administration Agent 0' is running.
'Tableau Server Administration Controller 0' is running.
'Tableau Server Service Manager 0' is running.
'Tableau Server License Manager 0' is running.
'Tableau Server Client File Service 0' is running.
'Tableau Server Database Maintenance 0' is stopped.
'Tableau Server Backup/Restore 0' is stopped.
'Tableau Server Site Import/Export 0' is stopped.
'Tableau Server SAML Service 0' is stopped.
Tableau Server Data Engine

Hyper is Tableau's in-memory Data Engine technology optimized for fast data ingests and analytical query processing on large or complex data sets. Starting in Tableau 10.5 release, Hyper powers the Data Engine in Tableau Server, Tableau Desktop, Tableau Online, and Tableau Public. The Data Engine is used when creating, refreshing or querying extracts. It is also used for cross-database joins to support federated data sources with multiple connections.

<table>
<thead>
<tr>
<th>Process</th>
<th>Data Engine</th>
</tr>
</thead>
<tbody>
<tr>
<td>Status</td>
<td>Status of the Data Engine process is visible on the Status Page. For more information, see View Server Process Status</td>
</tr>
<tr>
<td>Logging</td>
<td>Logs generated by the Data Engine process are located in <code>/var/opt/tableau/tableau_server/data/tabsvc/logs/hyper</code>. For more information, see Server Log File Locations</td>
</tr>
</tbody>
</table>

Memory and CPU usage

The Data Engine is designed to leverage all available CPU and memory on the machine to provide the fastest response times.

CPU usage

Hyper technology leverages the new instruction sets in CPU and is capable of parallelizing and scaling to all the available cores. Hyper technology is designed to scale to many cores efficiently, and also to maximize the use of each single core as much as possible. This means that you can expect to see the CPU being fully used during query processing. Adding more CPU is expected to result in performance improvement.

Modern operating systems such as Microsoft Windows, Apple macOS, and Linux have mechanisms to make sure that even if a CPU is fully used, incoming and other active processes can run simultaneously. In addition, to manage overall resource consumption and to prevent overloading and completely starving other processes running on the machine, the Data Engine monitors itself to stay within the limits set in the Tableau Server Resource Manager.
Tableau Server Resource Manager (SRM). Tableau Server Resource Manager monitors the resource consumption and notifies Data Engine to reduce the usage when it exceeds the predefined limit.

Since the Data Engine is designed to utilize the available CPU, it is normal to see spikes in CPU usage at times. If however, you see high CPU usage (e.g., 95%) for extended periods of time (an hour or more), this can mean a couple of things:

- There is a high load of queries. This can happen if a server is under stress due to overload of multiple client requests and the queries are queuing up. If this happens often, it is an indication that more hardware is required to serve the clients. Adding more CPU in this case should help to improve performance.

- There is one long running query. In this case, the Tableau Server resource Manager will stop long running queries based on the timeout settings. This was also true for the Tableau Server versions earlier than version 10.5

For more information on Tableau Server Resource Manager, see General Performance Guidelines.

Memory usage

Memory usage of the Data Engine depends on the amount of data required to answer the query. The Data Engine will try to run this in-memory first. A working set memory is allocated to store an intermediate data structure during query processing. In most cases, systems have enough memory to do these types of processing, but if there isn't enough available memory, or if more than 80% of RAM is utilized, the Data Engine shifts to spooling by temporarily writing to disk. The temporary file get deleted after the query has been answered. Therefore, spooling is an indication that more memory may be needed. Memory usage should be monitored and upgraded appropriately to avoid performance issues caused by spooling.

To manage memory resources on the machine, the maximum memory limit for Data Engine is set by Tableau Server Resource Manager (SRM).
Server configuration, Scalability, and Performance

- A single instance of Data Engine is automatically installed per node where an instance of File Store, Application Server (VizPortal), VizQLServer, Data Server, or Backgrounder is installed. The Data Engine can scale by itself and uses as much CPU and memory as needed, thus removing the need for multiple instances of the Data Engine. For more information on the server processes, see Tableau Server Processes.

- The instance of Data Engine installed on the node where File Store is installed is used for querying data for view requests. The instance of Data Engine installed on the node where backgrounder is installed is used for extract creation and refreshes. This is an important consideration when you are doing performance tuning. For more information, see Performance Tuning Examples.

- Data Server, VizQL Server, and the Application Server (VizPortal) all use the local instance of Data Engine to do cross-database joins and create shadow extracts. Shadow extract files are only created when you work with workbooks that are based on non-legacy Excel or text, or statistical files. Tableau creates a shadow extract file in order to load the data more quickly.

- In Tableau Server 10.5 one instance of Data Engine is installed automatically when you install backgrounder. The backgrounder process uses the single instance of Data Engine (hyperd.exe) installed on the same node.

Scalability:

You can scale up with the new Data Engine: Since cores are fully utilized, adding more cores makes individual queries execute faster which in turn allows for more queries to execute in less time.

Memory usage should be monitored and upgraded appropriately to avoid the performance issues caused by spooling.

For more information on Scalability, see Tableau Server Scalability.
Performance:

As mentioned in the beginning of this topic, in most cases you can expect to see performance improvements when you upgrade to Tableau 10.5 or later. However, in some cases you may see performance issues like slower response times, especially for workbooks with complex calculations. To test your workbooks to see if should expect performance issues, we recommend that you test this before you upgrade. For more information, see Test Workbook Performance. If you have already upgraded your Tableau Server, see Troubleshoot Performance of Workbooks with Calculations.

For more information on performance, start with General Performance Guidelines, and Performance Tuning

Performance benefits

Starting in 10.5, Hyper technology has been integrated with Tableau Data Engine to give you the following key benefits:

- **Faster extract creation**: With Hyper technology, extracts are generated almost as fast as the source system can deliver data, no sorting needed.

- **Support for larger extracts**: Prior to this release, you might have not been able to get all your data into a single extract. With Hyper technology, much larger amounts of data can be included in a single extract.

- **Faster analysis of extracts**: In many cases you will see faster querying of data for larger extracts, or workbooks with complex calculations.

Here are some reasons why the Data Engine powered by Hyper performs better on larger or complex extracts and is optimized for faster querying:

- **Hyper technology is designed to consume data faster**: Unlike in previous versions, the Data Engine does not do any post processing like sorting. With Hyper, post processing steps like sorting are not needed giving the Data Engine the ability to perform better with larger extracts.
Hyper technology is memory-optimized. This means that when needed, all data lives in memory. This results in fast data access times.

Hyper technology is CPU optimized. This means that Data Engine now fully parallelizes the query execution and utilizes available CPU in such a way that query execution time scales almost linearly with the number of cores in the machine.

Hyper is a compiling query engine. Queries are either interpreted or compiled to the machine code for maximum performance and allowing the Data Engine to get most performance out of modern hardware (CPU, large main-memory capacities).

Hyper technology uses advanced query optimizations to make queries faster. Along with many additional advanced techniques such as, materializing min and max values for each column, mini-indices to optimize search ranges, more granular data block-level dictionaries, advanced logic for join and sub-query performance optimizations, the new Data Engine offers many improvements over the previous Tableau Data Engine in terms of performance and scalability.

Tableau Server Data Server

The Data Server manages connections to Tableau Server data sources. To make Data Server highly available, configure one or more Data Server processes to run on multiple nodes of the cluster.

<table>
<thead>
<tr>
<th>Process</th>
<th>Data Server</th>
</tr>
</thead>
<tbody>
<tr>
<td>Status</td>
<td>Status of the Data Server process is visible on the Status Page. For more information, see View Server Process Status</td>
</tr>
<tr>
<td>Logging</td>
<td>Logs generated by the Data Server process are located in /var/-opt/tableau/tableau_server/data/tabsvc/logs/dataserver. For more information, see Server Log File Locations</td>
</tr>
</tbody>
</table>

What happens if a Data Server process fails? Queries running through the Data Server process will fail, resulting in a failed view rendering, extract refresh, or alert. Subsequent
requests, including a retry of the failed operation, should succeed as long as a working Data Server is available to accept rerouted requests.

Tableau Server is not dependent on Data Server to function; however, without a running Data Server, workbooks on the server lose the ability to query or to connect to published data sources. Any view that does not use Data Server for any of its data sources will still function correctly.

**Tableau Server File Store**

The Tableau Server File Store process controls the storage of extracts. When the file store is installed, an instance of the Data Engine is also installed unless the node already has an instance of the data engine. In highly available (HA) environments, the File Store ensures that extracts are synchronized to other file store nodes so they are available if one file store node stops running.

<table>
<thead>
<tr>
<th>Process</th>
<th>File Store</th>
</tr>
</thead>
<tbody>
<tr>
<td>Status</td>
<td>Status of the File Store process is visible on the Status Page. For more information, see View Server Process Status</td>
</tr>
<tr>
<td>Logging</td>
<td>Logs generated by the File Store process are located in /var/-opt/tableau/tableau_server/data/tabsvc/logs/filestore. For more information, see Server Log File Locations</td>
</tr>
</tbody>
</table>

The decommission Command

If you want or need to remove a file store you should decommission the file store first, using the `tsm topology filestore decommission` command. If you don't decommission the filestore before you attempt to remove it, you will be prompted to do so. Decommissioning puts the file store into read-only mode and copies any unique data contained in the file store to the other file store(s) in the cluster. While a file store is being decommissioned, this shows on the Status page, and once all unique content has been copied to other file store nodes, the decommissioned node shows as ready to be removed.
Tableau Server Gateway Process

The Tableau Server gateway process is an Apache web server component (httpd.exe). Its role is to handle requests to the server from all clients—Tableau Desktop, mobile devices, a proxy, a load balancer, etc.

The server runs a single instance of the gateway process; you can't run more than one per machine.

<table>
<thead>
<tr>
<th>Process</th>
<th>Gateway</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Status</strong></td>
<td>Status of the Gateway is visible on the Status Page. For more information, see View Server Process Status</td>
</tr>
<tr>
<td><strong>Logging</strong></td>
<td>Logs generated by the repository are located in /var/-opt/tableau/tableau_server/data/tabsvc/logs/httpd. For more information, see Server Log File Locations</td>
</tr>
</tbody>
</table>

Port assignment

By default, the gateway process listens for requests on port 80 (for HTTP requests) and 443 (for SSL requests). When you install Tableau Server on a computer, part of the server configuration makes sure that this port is open in the computer’s firewall. If the computer is running a different process that requires port 80 for HTTP, you can change the port assignment for the gateway process during installation. (You cannot change the SSL/HTTPS port.)

If you need to set the HTTP port after installation, then run the following TSM command, where node is the node number (for example, node1) and nn is the new port number:
If you are running Tableau Server with a reverse proxy server, you will need to configure other port and host-related settings on Tableau Server. See Configuring Proxies for Tableau Server.

Log files for the gateway process

The gateway process creates two sets of log files:

- Activity logs. The name for these log files has the format `access.yy_mm_dd_hh_mm_ss.log`.

- Error logs. All errors are logged in a single file named `error.log`.

For more information, see Log File Snapshots (Archive Logs).

Gateway processes in a cluster

If your server environment is distributed across multiple machines, you can run a single gateway process on each node of the cluster. The most common scenario for running a gateway process on multiple computers in the cluster is that you have a load balancer in front of the cluster. In this scenario, the load balancer distributes requests to any gateway in the cluster.

If you need to take a node off line (for example, to perform maintenance on that node), you can disable the load balancer's routing to that machine. When the maintenance is complete, you can re-enable the node on the load balancer.

You must have a gateway process running on at least one computer in the cluster. If you remove the gateway process from the primary server, you must make sure that another computer in the cluster is running the gateway process. You must also make sure that that computer is reachable by clients.

If the Tableau Server is configured to use SSL, you must make sure that the certificate for SSL support is in the same location on each computer in the cluster that has the gateway process running.
process running. For more information about using SSL, see Configure SSL for External HTTP Traffic to and from Tableau Server.

Similarly, if the server installation uses a custom logo, the logo must be in the same location on every computer that is running the gateway process.

If you need to change the port number that the gateway process listens on, as explained earlier, you can use the configuration dialog box or run the following command for each worker computer that is running the gateway process:

```
 tsm topology set-ports --node-name node --port-name gateway:primary --port-value nn
```

Additional information

Configuring Proxies for Tableau Server

Add a Load Balancer

**Tableau Server Messaging Service**

The Tableau Server messaging service uses Apache ActiveMQ. It is a publish/subscribe platform that enables secure, scalable, performant, and highly available message-oriented communication for microservices. The Messaging Service is used to support communication between microservices in Tableau Server.

The server runs a single instance of the Messaging Service.

<table>
<thead>
<tr>
<th>Process</th>
<th>Messaging Service</th>
</tr>
</thead>
<tbody>
<tr>
<td>Status</td>
<td>Status of the Messaging Service is visible on the Status Page and from the command line using the <code>tsm status -v</code> command. For more information, see View Server Process Status</td>
</tr>
<tr>
<td>Logging</td>
<td>Logs generated by the repository are located in <code>/var/opt/tableau/tableau_server/data/tabsvc/logs/activemqserver</code>. For more information, see Server Log File Locations</td>
</tr>
</tbody>
</table>
Impact if Messaging Service is not running properly

If the Messaging Service stops or fails to start the consequences are relatively mild. Tableau Server will still work but status will show as "Degraded", and users mentioned in comments will not receive email notification of the mention. There is no loss of data.

If the Messaging Service stops, it is automatically restarted; as long as the computer itself is otherwise healthy, the Messaging Service should restart. If you cannot restart the service, you can disable it so that messaging uses the older mechanism.

Disabling a failed Messaging Service instance

If the Messaging Service will not automatically restart:

1. Disable use of the service by using the TSM CLI to set these options to false and apply the pending changes:

   ```
   tsm configuration set -k features.MessageBusEnabled -v false
   tsm configuration set -k features.ActiveMQ -v false
   tsm pending-changes apply
   ```

2. Restart Tableau Server (if you did not restart the server when you applied pending changes).

These steps configure Tableau Server to use the pre-2019.4 messaging mechanism.

Messaging Service in a cluster

You must run one instance of the Messaging Service in a Tableau Server installation. If your server is distributed across multiple node, you can run an instance of the Messaging Service on any one node in the cluster. Only one instance of the service can be configured.

Tableau Server Microservice Containers

The two Microservice Container processes, Interactive Microservice Container and Non-Interactive Microservice Container, are added automatically to a Tableau Server node.
when a microservice contained in one of these container processes is added to the server. The microservices within these container processes are internal to Tableau Server. While you cannot add a Microservice container directly, you can change the number of instances for both Microservice Containers, when necessary.

It is strongly recommended that you do not change the number of Microservice Container instances unless specifically recommended to do so. Following are scenarios where you may need to configure the number of Microservice Containers:

- Enable Tableau Catalog: To improve the time taken for the Tableau Catalog indexing process, you may need to configure the number of threads used by the process. For more details, see Enable Tableau Catalog.

Microservice Containers and the microservices:

- Interactive Microservice Container:
  - Relationship Query Microservice
  - Content Model Microservice

- Non-Interactive Microservice Container:
  - Relationship Ingestor Microservice
  - Content Provider Microservice

Viewing Microservice Container Status

You can see the status of the Microservice Container processes from the TSM Status page, or from the command line, using the `tsm status -v` command. When you use the TSM Status page to View Server Process Status, the status of each container process is visible, but you cannot see the status of any of the microservices in the containers. When you use the command line, more detail is shown, including the status of each individual microservice.

Microservice Container Status

The status of a container process depends on the status of the microservices within the container. When all microservices within a container process are running as expected, the
container status is Active (on the TSM Status page) or running (when viewed from the TSM command line). If all microservices within a container process are stopped, the status for the container is Error (on the TSM Status page) or stopped (from the TSM command line). If a microservice is stopped but at least one other microservice is running, the container status is Degraded (on the TSM Status page) or degraded (from the TSM command line).

When all microservices within a container process have a status of running, the container status is Active. If any microservice in a container is in an error state (has a status of stopped) the container process status is degraded. If all microservices in a container are in an error state, the container status is error.

Use the TSM web interface

To view the Microservice Container status from the TSM Status page:

1. Open TSM in a browser:

   http://<tsm-computer-name>:8850

2. Click Status:

   The page displays the status for the Interactive Microservice Container and Non-Interactive Microservice Container processes, as well as status for other processes running as part of TSM or Tableau Server.

   You cannot see the status of any individual microservice within a container process, but if the container process has a status of Active (a green check), the microservices it contains are all running as expected. To see the status of individual microservices, use the TSM command line.

**Note:** The status of the container processes does not display on the older Tableau Server status page. For details about the two status pages and how they differ, see View Server Process Status.
Use the TSM CLI

To view the Microservice Container status from the TSM command line:

1. Open a command prompt as administrator on the initial node (the node where TSM is installed).

2. Run the following command:

   tsm status -v

The output from the command shows you the status of the container services and the microservices in them:

node1: TABLEAUSVR01
Status: RUNNING
'Tableau Server Gateway 0' is running.
'Tableau Server Application Server 0' is running.
'Tableau Server Interactive Microservice Container 0' is running.
'Tableau Server Cluster Controller 0' is running.
'Tableau Server Coordination Service 0' is running.
'Tableau Server Search And Browse 0' is running.
'Tableau Server Backgrounder 0' is running.
'Tableau Server Backgrounder 1' is running.
'Tableau Server Non-Interactive Microservice Container 0' is running.
'Relationship Ingestor Microservice 0' is running.
'Content Provider Microservice 0' is running.
'Tableau Server Data Server 0' is running.
'Tableau Server Data Server 1' is running.
'Tableau Server Data Engine 0' is running.
'Tableau Server File Store 0' is running.
'Tableau Server Repository 0' is running (Active Repository).
'Tableau Server Tableau Prep Conductor 0' is running.
'Tableau Server Elastic Server 0' is running.
'Tableau Server Ask Data 0' is running.
'Tableau Server Administration Agent 0' is running.
'Tableau Server Administration Controller 0' is running.
'Tableau Server Service Manager 0' is running.
'Tableau Server License Manager 0' is running.
'Tableau Server Client File Service 0' is running.
'Tableau Server Database Maintenance 0' is stopped.
'Tableau Server Backup/Restore 0' is stopped.
'Tableau Server Site Import/Export 0' is stopped.
'Tableau Server SAML Service 0' is stopped.
c:\Program Files\Tableau\Tableau Server-\packages\scripts.near.18.1216.1859>

Tableau Server Repository

Tableau Server Repository is a database that stores server data. This data includes information about Tableau Server users, groups and group assignments, permissions, projects, data sources, and extract metadata and refresh information.

The Repository is also referred to as PostgreSQL.

<table>
<thead>
<tr>
<th>Process</th>
<th>Repository</th>
</tr>
</thead>
<tbody>
<tr>
<td>Status</td>
<td>Status of the Repository is visible on the Status Page. For more information, see View Server Process Status</td>
</tr>
</tbody>
</table>
Logging

Logs generated by the repository are located in /var/opt/tableau/tableau_server/data/tabsvc/logs/pgsql. For more information, see Server Log File Locations.

Preferred active repository

When you configure Tableau Server you have the option to specify a node as the preferred active repository. When Tableau Server is configured for repository failover, the preferred active repository node is the one used for the active repository. This is an optional step, and if you do not specify a preferred active repository node, Tableau Server will select the active repository node on startup.

To configure the preferred active repository, use the tsm configuration set command to configure the `pgsql.preferred_host` option:

```
tsm configuration set -k pgsql.preferred_host -v "<host_name>"
```

**Note:** The `host_name` is case-sensitive and must match the node name shown in the output of `tsm status -v`.

Configure a preferred active repository node if you want Tableau Server to select a specific node on startup. You might want to do this if you have a particular server you want to use for your active repository (a computer with more disk space or memory for example), or if you are using custom administrative views. Custom administrative views have embedded connection information that refers to the repository for which you created the views. For more information on connecting to the Tableau Server repository, see Collect Data with the Tableau Server Repository.

**The failoverrepository Command**

If failover occurs and your passive repository becomes the active repository, it remains the active repository until either Tableau Server restarts or you use the `tsm topology failover-repository` command to switch back. Specify the repository you want to be the
active one, or specify that the preferred active repository (if configured) should be made active again. For more information, see tsm topology failover-repository.

Tableau Server SAML Service

For Tableau Server installations that have site-specific SAML enabled, there will also be a SAML Service instance running on each node that is configured with Application Server. This will be automatically configured when site-specific SAML has been enabled on the server. SAML Service on Tableau Server will show as stopped unless site SAML is enabled.

<table>
<thead>
<tr>
<th>Process</th>
<th>SAML Service</th>
</tr>
</thead>
<tbody>
<tr>
<td>Status</td>
<td>Status of the SAML Service process is not visible on the Status Page. Use the TSM CLI to view status. For more information, see View Server Process Status</td>
</tr>
<tr>
<td>Logging</td>
<td>Logs generated by the SAML Service process are located in /var/-opt/tableau/tableau_server/data/tabsvc/logs/samlservice. For more information, see Server Log File Locations</td>
</tr>
</tbody>
</table>

If this process goes into a failed state, then users will not be able to sign in to Tableau Server if the user request is routed to an Application Server on this node. Like other processes, when a node containing the SAML Service fails, the failed process should automatically restart within seconds.

Tableau Server Search and Browse

The Search & Browse process handles fast search, filter, retrieval, and display of content metadata on the server. To configure high availability for the Search & Browse process, configure the process on multiple nodes.
What happens if a Search & Browse process fails? Users can still sign in to Tableau Server, but workbook content will appear to be missing. The content is not actually missing. Rather, the content is not being returned in the search results; it will be displayed again after the Search and Browse process restarts.

If more than one Search & Browse process is configured and running on multiple nodes when the failure occurs, requests made to a failed Search & Browse process will also fail, but subsequent requests will be routed to working Search & Browse processes. Each Search & Browse process indexes across all nodes in the cluster, therefore if all but one Search & Browse process fails, results will still be returned across all nodes.

Tableau Server Administration Controller Process

The Administration Controller process hosts the TSM REST API for configuring and managing your Tableau Server deployment. There can only be a single instance of the Administration Controller in the entire cluster.

The Administration Controller is also referred to as the **TSM Controller**.
If the Administration Controller fails, the Tableau Server cluster should continue to function; however, you will not be able to make any changes or updates to the configuration or topology until the Administration Controller is back up and running. Like other TSM services, Administration Controller is automatically restarted if it is stopped or has failed.

If automatic restart continues to fail, move the Administration Controller process to another node in the cluster to mitigate the situation. See Recover from an Initial Node Failure.

To manually start the Administration Controller Manager, run the following commands:

```bash
sudo su -l tableau
systemctl --user start tabadmincontroller_0
systemctl --user status tabadmincontroller_0
```

If the Administration Controller is not started, then run,

```bash
systemctl --user restart tabadmincontroller_0
```

### Tableau Server TSM Maintenance Services

There are three TSM Maintenance Services that are installed on every node of the cluster: Database Maintenance, Backup/Restore, and Site Import/Export.

<table>
<thead>
<tr>
<th>Processes</th>
<th>Database Maintenance, Backup/Restore, and Site Import/Export.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Status</td>
<td>Status of the TSM Maintenance services are not visible on the Status Page. Use the TSM CLI to view status. For more information, see View Server Process Status.</td>
</tr>
<tr>
<td>Logging</td>
<td>Logs generated by the Service Manager process are located in /var/opt/tableau/tableau_server/data/tabsvc/logs/, in the databasesmaintenance,backuprestore, and siteimportexport directories. For more information, see Server Log File Locations.</td>
</tr>
</tbody>
</table>
These services remain stopped unless a specific maintenance task that requires them is initiated by the administrator. Additional high-availability configuration is not required for these services. These services are used only for maintenance tasks such as backup and restore and should not impact the functioning of Tableau Server for the end users.

**Tableau Server VizQL Server**

The VizQL Server loads and renders views, and computes and executes queries. To achieve high availability for the VizQL Server process, configure one or more instances to run on multiple nodes.

<table>
<thead>
<tr>
<th>Process</th>
<th>VizQL Server</th>
</tr>
</thead>
<tbody>
<tr>
<td>Status</td>
<td>Status of the VizQL Server process is visible on the Status Page. For more information, see View Server Process Status</td>
</tr>
<tr>
<td>Logging</td>
<td>Logs generated by the VizQL Server process are located in <code>/var/opt/tableau/tableau_server/data/tabsvc/logs/vizqlserver</code>. For more information, see Server Log File Locations</td>
</tr>
</tbody>
</table>

What happens if a VizQL Server process fails? If there is only one VizQL Server process and it fails, then Tableau Server will no longer be able to render any views. High availability requires configuring redundant VizQL processes. A fairly typical configuration consists of two to four VizQL Server processes on each node.

This simultaneously serves the need for high availability and scalability. If multiple VizQL Server processes are running, then the failure of a single process will result in the failure of any requests and the loss of session data at the time of its failure. Any future requests will be routed to the other working VizQL Server processes on the Tableau Server cluster.
Tableau Prep Conductor

Overview

If you build flows in Tableau Prep, you can automate running your flows to refresh flow input and output data on a schedule using Tableau Server, instead of opening Tableau Prep Builder to run individual flows manually.

Tableau Prep Conductor is one of the processes on Tableau Server. It runs flows, checks connection credentials, and sends alerts if a flow fails. Tableau Prep Conductor leverages the scheduling and tracking functionality of Tableau Server so you can automate running flows to update the flow output instead of logging into Tableau Prep Builder to manually run individual flows as your data changes.

Tableau Prep Conductor is licensed through the Data Management Add-on. To use Tableau Prep Conductor, you must have the Data Management product key activated on your server. For more information, see Licensing Tableau Prep Conductor.

Tableau Prep Conductor uses the following components to run flows:

- **Backgrounder**: Tableau Prep Conductor uses the Backgrounder process to run flows. Backgrounder is single threaded, so each instance of the Backgrounder process on a node can run one flow at a time. By adding more Backgrounders to a node, you can increase the number of flows that can be run in parallel on that node. The Backgrounder processes can be up to half the number of the physical cores of that node.

- **Connectors**: Prep Conductor uses the supported Tableau Data connectors to connect to data. For a list of supported Connectors, see Supported Connectors.

- **Data Engine**: Any changes to data or transformation steps in your flow that cannot be pushed to the underlying data source are processed using the Data Engine process. For example, SQL Server does not natively support regular expressions. When connecting to SQL Server, Tableau Prep lets you write regular expression cal-
Tableau Prep Conductor uses Data Engine to temporarily load the data and then perform the regular expression.

Performance and Scale Recommendations

- **Isolate flows to a separate node**: Running Tableau Prep Conductor on a separate node will isolate flow workflows from other Tableau workloads. This is highly recommended since Prep flows are CPU and RAM intensive.

- **Manage flow schedules**: You can control flow execution by creating flow schedules. These schedules let you determine when flows run, how frequently they run, the priority of that schedule, and whether to run items in that schedule serially or in parallel.

- **Add resources**: When scaling your Tableau Prep Conductor environment, we recommend scaling up to an 8 physical cores box per node running as many as 4 back-enders on each. As you need more resources, we recommend adding more nodes to your server environment.

You can monitor user activity and performance of flows using Administrative views. For more information, see Monitor Flow Health and Performance.

Topology and Configuration

By default, the installer enables Tableau Prep Conductor on all nodes that already have back-grounder enabled and where the following conditions are true:

- The Tableau Server has a valid Data Management product key.

- The node role is either set to allow all tasks or to allow only flows. For more information, on node roles, see the Node Roles in Server topic.

It is recommended that you enable Tableau Prep Conductor on a dedicated node to run flows. For more information, see the following topics:

- For new Tableau Server installations, see Installing Tableau Server with Prep Conductor.
To enable Tableau Prep Conductor on an existing installation of Tableau Server, see Enable Tableau Prep Conductor to your Tableau Server Installation topic.

Enabling Tableau Prep Conductor on Tableau Server

To enable Tableau Prep Conductor, use the Tableau Data Management product key. Tableau Prep Conductor is now licensed and enabled at the server level.

Before you can start publishing flows to your Tableau Server, there are server-level and site-level settings you must configure or verify to prepare your Tableau Server to allow publishing, scheduling and monitoring flows.

Review the following topics to understand Tableau Prep Conductor licensing, and learn how to enable Tableau Prep Conductor:

- Enable and Configure Tableau Prep Conductor on Tableau Server: This topic provides step-by-step instructions on how to enable and configure Tableau Prep Conductor, and preparing your Tableau Server to allow publishing, scheduling, managing, and maintaining flows.

- License Overview: This topic describes licensing.

Tableau Server Dynamic Topology Changes

With the introduction of TSM, Tableau Server also introduced the ability to make certain topology changes or updates without restarting the server. These are known as dynamic topology changes, and are possible with the backgrounder and VizQL Server processes.

You can increase or decrease the number of backgrounder or VizQL Server instances on a node that already has at least one of these processes running, without any Tableau Server downtime. If you have a node already running backgrounder, you add additional instances of backgrounder or VizQL Server to the node without restarting Tableau. You can do the same with VizQL Server node running VizQL Server, adding or removing instances of either backgrounder or VizQL Server. You must be only changing the number of instances
of backgrounder or VizQL Server. If you also add or remove another process, you need to restart Tableau Server. For example, if you have a node running four instances of VizQL Server, you can remove all those instances and add one or more instances of backgrounder instead, without restarting Tableau Server, taking advantage of dynamic topology changes.

Example Scenarios

To better understand why this might be useful, consider these examples:

- **Backgrounder**—At the end of a sales quarter your sales team is using Tableau Server to keep track of their numbers. Dashboards that depend on extracts are showing sales people how they are doing. Any delay in extract refreshes means your team is not seeing the most up-to-date numbers. You can add additional backgrounders to any node that already has at least one backgrounder or VizQL Server, and increase the throughput of extract refreshes, helping to guarantee the numbers are up-to-date as your team finished up their quarter. Later, after the quarterly push, you can reduce the backgrounder instances again to return Tableau Server to its original configuration.

- **VizQL Server**—Similarly, if Tableau Server is unable to keep up with view refreshes, you can quickly add additional VizQL Server instances to any node that already has at least one instance of either VizQL Server or backgrounder configured. In the above backgrounder example, you might want to remove VizQL Servers temporarily, to accommodate additional backgrounders, and then re-add them back before your users arrive in the morning.

Making dynamic topology changes

You can make dynamic topology changes using the TSM Web UI, or on the command line. To use the Web UI, sign in to TSM using a browser, and on the Configuration tab, update the number of backgrounder or VizQL Server instances for the node you are updating. For details, see Configure Nodes. To make your changes using the TSM CLI, at a command prompt, run the tsm topology set-process command. For details, see Changing the number of processes on a node.

Impact of dynamic topology changes

When making dynamic topology changes that remove existing instances of VizQL Server or backgrounder, the instances are removed immediately. Be aware of the following potential
impacts to users and currently running jobs:

- **Backgrounder**—Any currently running jobs are terminated. The normal Tableau Server retry logic will restart these jobs, using another backgrounder instance.

- **VizQL Server**—Any currently active sessions are terminated. Users may see an error message. Refreshing the browser should clear the error.

**Best practices**

Tableau recommends you test any dynamic topology changes you plan on using, before implementing them in your production environment. This will help you fully understand potential impacts to your users and scheduled refreshes and subscriptions, and allow you to most efficiently take advantage of the flexibility offered by dynamic topology.

**Automating dynamic topology changes**

You can automate dynamic topology changes. For example, if you have most of your extract refreshes scheduled overnight, and know your server has extra capacity because users are not signed in, you can use a script or other automated deployment tool to add backgrounder instances when they can be most efficiently used, and then remove them before the start of the work day.

To automatically get the status of processes, use the `tsm status -v` command and parse the output in your script. Alternately, you can use the TSM REST API get server status. The API is currently in alpha. For more details, see [Get server status](https://example.com).

A sample script to set four instances of backgrounder on node2 might look like this:

```bash
echo Adding/Removing Processes
tsm topology set-process -pr backgrounder -n node2 -c 4
tsm pending-changes apply
echo Done!
```

**Additional information**

**Tableau Server Processes**
Server Process Limits

When you reconfigure processes for Tableau Server, there is a limit to the amount that you can increase the number of process instances. By default, the limit is set to eight. If your machine has enough RAM and CPU cores, and you want to go above this limit, you can change the limit using the `service.max_procs` configuration option. For each process instance, Tableau recommends that the machine running the process have at least 1 GB of RAM and 1 logical CPU core.

To change the maximum number of processes allowed:

1. Type the following command, where `number` is the maximum number of process instances you want to allow:

   ```bash
   tsm configuration set -k service.max_procs -v <number>
   ```

   For example:

   ```bash
   tsm configuration set -k service.max_procs -v 10
   ```

2. Next type:

   ```bash
   tsm pending-changes apply
   ```

   The `pending-changes apply` command displays a prompt to let you know this will restart Tableau Server if the server is running. The prompt displays even if the server is stopped, but in that case there is no restart. You can suppress the prompt using the `--ignore-prompt` option, but this does not change the restart behavior.

   For more information, see `tsm pending-changes apply`.

Tableau Services Manager Ports

The processes and services that make up the components of Tableau Services Manager (TSM) and Tableau Server on Linux use various ports to communicate. By default, most these ports are assigned (mapped) dynamically from a predefined range of ports. The port
assignments are made for each service or process when it is installed. You can control what ports are used in a variety of ways.

A small subset of processes do not use dynamic port mapping and behave uniquely. For more information, see Ports that are not dynamically mapped, below.

**Note:** This operation includes steps that you may need to perform using the TSM command line.

**Firewall requirements**

If you are running a firewall on the computer where you will be installing Tableau Server, then you will need to open the following default ports for Tableau Server traffic. All port numbers, except 443 can be changed.

<table>
<thead>
<tr>
<th>Port</th>
<th>TCP/UDP</th>
<th>Used by ...</th>
<th>TYPE OF INSTALLATION</th>
</tr>
</thead>
<tbody>
<tr>
<td>80</td>
<td>TCP</td>
<td>Gateway</td>
<td>All</td>
</tr>
<tr>
<td>443</td>
<td>TCP</td>
<td>SSL. When Tableau Server is configured for SSL, the application server redirects requests to this port. Do not change this port.</td>
<td>X</td>
</tr>
<tr>
<td>8850</td>
<td>TCP</td>
<td>Tableau Services Manager.</td>
<td>X</td>
</tr>
<tr>
<td>8060</td>
<td>TCP</td>
<td>PostgreSQL database.</td>
<td>X</td>
</tr>
<tr>
<td>8061</td>
<td>TCP</td>
<td>PostgreSQL backup verification port</td>
<td>X</td>
</tr>
<tr>
<td>8000-9000</td>
<td>TCP</td>
<td>Range of ports reserved by default for dynamic mapping of Tableau processes</td>
<td>X</td>
</tr>
</tbody>
</table>
Tableau Server on Linux Administrator Guide

<table>
<thead>
<tr>
<th>Port</th>
<th>TCP/UDP</th>
<th>Used by ...</th>
<th>TYPE OF INSTALLATION</th>
</tr>
</thead>
<tbody>
<tr>
<td>27000-27009</td>
<td>TCP</td>
<td>Range of ports used by Tableau Server for License service. This range must be open on the node running the License service and accessible from other nodes. By default, the initial node runs the License service.</td>
<td>X Distributed / High Availability</td>
</tr>
</tbody>
</table>

See Configure Local Firewall.

Port assignment

There are two approaches you can use for port assignment or mapping in TSM:

- **Dynamic port assignment.** This is the default and requires the least intervention by the administrator.
- **Manual port assignment.** This option requires an administrator to individually assign each port.

If you change port assignments and you are running a local firewall, see Local firewall configuration.

Dynamic port assignment

You can control which ports are used by changing the range of ports available to the dynamic assignment process. Doing this leaves dynamic port assignment in place but restricts the ports that can be chosen. With dynamic port assignment in place, you can still choose to assign ports for certain processes manually. This approach enables you to assign specific ports to specific processes, while leaving the others to be mapped dynamically. Using dynamic mapping, with or without some individually assigned ports is the easiest approach,
and should satisfy the requirements of most customers. Dynamically assigned ports are preserved if you export your Tableau Server configuration.

By default, ports are assigned for each service or process from available ports between 8000 to 9000. This assignment takes place when services are installed for the first time on a node. After Tableau Server is initialized, you can see which ports are being used by which services or processes by running this command:

```
 tsm topology list-ports
```

If you have a multi-node cluster, ports on all nodes are listed.

Changing the port range

For organizations that have specific requirements for ports being used, the easiest way to control this is to change the range from which ports are dynamically selected. You need to do this at installation, by specifying a minimum and maximum port for the range in your configuration file.

```
   "configKeys": {
      "ports.range.min": "8300",
      "ports.range.max": "8600"
   }
```

**Note:** The minimum allowable size of your port range will depend on your server installation and how many services or processes you are running. As a general best practice you should not restrict the range too tightly because port assignment is done by selecting random ports within the range, and if you do not allow a large enough range, selection may fail to find an available port.

To limit the range from which available ports are chosen to those between 8300 and 8600, your configuration file would include an entry similar to this:
Disabling dynamic port assignment

If you need more control of port assignment than you can get through a combination of restricting port range and individually assigning ports, you can disable dynamic port mapping at initial server configuration. Disabling dynamic port mapping requires you to manually assign every port for every process, so we don’t recommend this unless you need to control every single port assignment.

If you disable dynamic port mapping, you must configure the port for each process on each node of your installation.

To disable dynamic mapping, your configuration file would include an entry similar to this:

```
"configKeys": {
  "service.port_remapping.enabled": false
}
```

**Important:** When you disable dynamic port assignments, the License service port range is not included. This range (27000-27009) must be open on the node running the License service and accessible from other nodes. By default, the initial node runs the License service.

Manual port assignment

You can disable automatic port assignment entirely and assign a port for each process individually. If you do this, you must assign a port for every process on every node. You can assign ports either in a configuration file, when the processes are first installed, or after installation, using a TSM command. Only ports assigned at process installation are preserved if you export your Tableau Server configuration.

You can specify individual ports for specific processes, whether or not dynamic mapping is enabled. You might do this if you want a process to use a particular port, or if you've disabled dynamic dynamic mapping. There are two ways to specify ports for processes: during installation or after installation.
Configuring ports during installation

We recommend configuring port assignment during the installation process as described here. Changing ports after installation is a much more labor-intensive process.

To configure ports during installation, create a JSON file that specifies your port configuration. This process is similar to defining a non-default port range, but instead you specify a particular port for a specific service or process. If you are going to assign specific ports, this approach is the most robust way to do so because the port mapping is preserved if you export the server configuration and topology settings using the `tsm settings export` command.

To define ports at installation, add information to your configuration file to specify the node (workerN), process (servicename) and instance ID (instanceid), port type (port-type), and the port to be used. The format looks like this:

`workerN.{servicename}_{instanceid}.{porttype}.port:X`

Where:

- `workerN` is an optional parameter and identifies the node for which the remapping applies. Node numbers start with zero (0). We recommend you do not include this parameter unless you need to map different ports for the same service on different nodes. If you leave this parameter off, you can map a service port on the initial node, or map the same service port on multiple nodes.
- `servicename` is the name of the process or service that will use the port.
- `instanceid` is the instance of the process. If you are going to be configuring multiple instances of a process on one node, you would need to increment this value for each instance. Start the instanceid at zero (0) and increment it by one (1) for each instance of the process. For services that only install a single instance on any given node, this must be left off.
- `porttype` If setting the primary port, do not include this option.
- `port` is the port the process or service should use.

For example, to set the port for the first instance of the file store process on the initial node to 8500, you would include a configuration file entry similar to this:
"configKeys": {
    "filestore_0.port": "8500"
}

The example above does not include the optional workerN parameter, so sets the port on all nodes in the cluster. It also leaves off the porttype option because it is setting the primary filestore port.

**Important:** When specifying port changes with a configuration file, you must include the --force-keys parameter with the `tsm settings import` command.

Configuring ports after installation

If you need to change ports after you have installed Tableau Server, use the `tsm topology set-ports` command. This approach allows you to specify a port for a specific process after that process has been installed. You are restricted by these limits:

- You must set ports individually, on each node.
- After you set an individual port, you must run `tsm restart`.
- The port assignments are not preserved if you need to import a Tableau Server configuration using `tsm settings import`.
- Port names use a different syntax for `tsm` commands than the syntax that is required for `configKeys`. The table at the end of this topic provides a syntax reference.

For example, to set second instance of the file store on the initial node to use port 8500:

```
  tsm topology set-ports --node-name node1 --port-name filestore -
  -port-value 8500 --instance 2
```

The following example shows how to use shorthand commands to set the JMX ports:

```
  tsm topology set-ports -n node1 --port-name vizqlserver:jmx.rmi
  -pv 9403 -i 1

  tsm topology set-ports -n node1 --port-name vizqlserver:jmx -pv
  9404 -i 2
```
Note: Port entries are not validated when you enter them. Therefore, if you use a port that is already assigned, or if you mistype the syntax for a command, Tableau will not give an error until you restart. After restarting, you may see a generic error, *The reconfigure async job failed.*

If you add an incorrect portname:type with a valid port, you cannot delete the incorrect entry. To update the port, you must reassign an unused port to that value to free up the port again.

Ports that are not dynamically mapped

The Tableau Server repository uses two ports that are not dynamically mapped. These each have a default port that you can override using the `tsm configuration set` command and a process-specific parameter.

<table>
<thead>
<tr>
<th>Port names</th>
<th>Port (default)</th>
<th>Description</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><code>pgsql.port</code></td>
<td>8060</td>
<td>Port for the Tableau Repository (PostgreSQL database). To override this port: <code>tsm configuration set -k psql.port &lt;port&gt;</code></td>
<td></td>
</tr>
<tr>
<td><code>pgsql.verify_restore.port</code></td>
<td>8061</td>
<td>Port for verifying the integrity of a repository backup. To override this port: <code>tsm configuration set -k psql.verify_restore.port &lt;port&gt;</code></td>
<td></td>
</tr>
</tbody>
</table>
Because these ports do not use the dynamic port mapping system, they do not show up in the output of the `tsm topology list-ports` command. To see the value of these you need to use the `tsm configuration get -k <config.value>` command. For example:

`tsm configuration get -k pgsql.port`

**Controlling port remapping with initialize-tsm**

Port assignments are made when services are installed. This means that in order to manually map ports for the TSM-specific processes, you need to assign the ports when you run the initialize-tsm script. The script includes options to specify ports for individual TSM services, as well as options for defining the minimum and maximum of the port range used with dynamic mapping, and you can disable dynamic mapping.

The table below lists the options for ports when running the initialize-tsm script.

**Table: initialize-tsm script port options**

<table>
<thead>
<tr>
<th>Script option</th>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-i</td>
<td>&lt;port&gt;</td>
<td>Sets the Coordination Service client port.</td>
</tr>
<tr>
<td>-e</td>
<td>&lt;port&gt;</td>
<td>Sets the Coordination Service peer port.</td>
</tr>
<tr>
<td>-m</td>
<td>&lt;port&gt;</td>
<td>Sets the Coordination Service leader port.</td>
</tr>
<tr>
<td>-n</td>
<td>&lt;port&gt;</td>
<td>Sets the TSM agent file transfer port.</td>
</tr>
<tr>
<td>-o</td>
<td>&lt;port&gt;</td>
<td>Sets the TSM Controller port.</td>
</tr>
<tr>
<td>-l</td>
<td>&lt;min-port&gt;</td>
<td>Sets the bottom of the port range used</td>
</tr>
<tr>
<td>Script option</td>
<td>Parameter</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>-----------</td>
<td>-------------</td>
</tr>
<tr>
<td>-r</td>
<td>&lt;max-port&gt;</td>
<td>Sets the top of the port range used for dynamically mapping ports.</td>
</tr>
<tr>
<td>--disable-port-remapping</td>
<td></td>
<td>Disables dynamic port mapping. If you do this you must assign ports for every service or process used by TSM and Tableau Server. For more information, see Manual port assignment above.</td>
</tr>
</tbody>
</table>

Dynamically mapped ports

This table lists the processes or services that use dynamically mapped ports.

<table>
<thead>
<tr>
<th>Port names: syntax for json file (configKeys)</th>
<th>Port names: syntax for tsm CLI</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>appzookeeper_0.client.port</td>
<td>appzookeeper:client</td>
<td>Coordination Service client port.</td>
</tr>
<tr>
<td>appzookeeper_0.peer.port</td>
<td>appzookeeper:peer</td>
<td>Coordination Service peer port.</td>
</tr>
<tr>
<td>appzookeeper_0.leader.port</td>
<td>appzookeeper:leader</td>
<td>Coordination Service leader port.</td>
</tr>
<tr>
<td>backgrounder_0.port</td>
<td>backgrounder</td>
<td>Backgrounder primary port.</td>
</tr>
<tr>
<td>backgrounder_0.debug.port</td>
<td>backgrounder:debug</td>
<td>Backgrounder debug port.</td>
</tr>
<tr>
<td>Port names: syntax for json file (configKeys)</td>
<td>Port names: syntax for tsm CLI</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------------------------</td>
<td>--------------------------------</td>
<td>-------------</td>
</tr>
<tr>
<td>backgrounder_0.jmx.port</td>
<td>backgrounder:jmx</td>
<td>Backgrounder jmx port.</td>
</tr>
<tr>
<td>backgrounder_0.jmx.rmi.port</td>
<td>backgrounder:jmx.rmi</td>
<td>Backgrounder jmx rmi port.</td>
</tr>
<tr>
<td>backgrounder_0.recommendations.trainer.port</td>
<td>backgrounder:recommendations.trainer</td>
<td>Backgrounder recommendations port.</td>
</tr>
<tr>
<td>backuprestore.port</td>
<td>backuprestore</td>
<td>Backup/Restore service port.</td>
</tr>
<tr>
<td>cacheserver_0.port</td>
<td>cacheserver</td>
<td>Cache server port.</td>
</tr>
<tr>
<td>clustercontroller.status.port</td>
<td>clustercontroller:status</td>
<td>Cluster Controller status port.</td>
</tr>
<tr>
<td>clustercontroller.storage.port</td>
<td>clustercontroller:storage</td>
<td>Cluster Controller storage port.</td>
</tr>
<tr>
<td>databasemaintenance.port</td>
<td>databasemaintenance</td>
<td>Database Maintenance port.</td>
</tr>
<tr>
<td>dataserver_0.port</td>
<td>dataserver</td>
<td>Data server primary port.</td>
</tr>
<tr>
<td>dataserver_0.debug.port</td>
<td>dataserver:debug</td>
<td>Data server debug port.</td>
</tr>
<tr>
<td>dataserver_0.jmx.port</td>
<td>dataserver:jmx</td>
<td>Data server jmx port.</td>
</tr>
<tr>
<td>Port names: syntax for json file (configKeys)</td>
<td>Port names: syntax for tsm CLI</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------------------------</td>
<td>--------------------------------</td>
<td>-------------</td>
</tr>
<tr>
<td>dataserver_0.jmx.rmi.port</td>
<td>dataserver:jmx.rmi</td>
<td>Data server jmx rmi port.</td>
</tr>
<tr>
<td>filestore.port</td>
<td>filestore</td>
<td>File store primary port.</td>
</tr>
<tr>
<td>filestore.status.port</td>
<td>filestore:status</td>
<td>File Store status port.</td>
</tr>
<tr>
<td>gateway.port</td>
<td>gateway</td>
<td>Gateway port. This defaults to 80, and if that is not available, to 8080. If that is not available, it tries 8000. That sequence is followed whether or not dynamic port assignment is enabled or not. If none of those ports are available and dynamic mapping is enabled, it takes an available port within the defined range. The gateway port must be the same on all nodes in a</td>
</tr>
<tr>
<td>Port names: syntax for json file (configKeys)</td>
<td>Port names: syntax for tsm CLI</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------------------------</td>
<td>---------------------------------</td>
<td>-------------</td>
</tr>
<tr>
<td></td>
<td></td>
<td>multi-node cluster, so if port 80 is selected on the initial node this is the port that will be used on all nodes and if it is unavailable on one of the other nodes, gateway port selection will fail.</td>
</tr>
<tr>
<td>hyper.port</td>
<td>hyper</td>
<td>Data engine primary port.</td>
</tr>
<tr>
<td>hyper.connection.port</td>
<td>hyper:connection</td>
<td>Data engine connection port.</td>
</tr>
<tr>
<td>licenseservice.vendor_daemon.port</td>
<td>licenseservice:vendor_daemon</td>
<td>License service vendor daemon port. Used for licensing-related communications between nodes in a multi-node installation.</td>
</tr>
<tr>
<td>samlservice.port</td>
<td>samlservice</td>
<td>SAML service port.</td>
</tr>
<tr>
<td>searchserver.port</td>
<td>searchserver</td>
<td>Search server primary port.</td>
</tr>
<tr>
<td>searchserver.debug.port</td>
<td>searchserver:debug</td>
<td>Search server</td>
</tr>
<tr>
<td>Port names: syntax for json file (configKeys)</td>
<td>Port names: syntax for tsm CLI</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------------------------</td>
<td>--------------------------------</td>
<td>-------------</td>
</tr>
<tr>
<td>searchserver.jmx.port</td>
<td>searchserver:jmx</td>
<td>debug port.</td>
</tr>
<tr>
<td>searchserver.jmx.rmi.port</td>
<td>searchserver:jmx:rmi</td>
<td>Search server jmx rmi port.</td>
</tr>
<tr>
<td>searchserver.startup.port</td>
<td>searchserver:startup</td>
<td>Search server startup port.</td>
</tr>
<tr>
<td>siteimportexport.port</td>
<td>siteimportexport</td>
<td>Site Import/Export port.</td>
</tr>
<tr>
<td>tabadmincontroller.port</td>
<td>tabadmincontroller</td>
<td>TSM Controller port.</td>
</tr>
<tr>
<td>tabadminagent.columbo.port</td>
<td>tabadminagent:columbo</td>
<td>Administration Agent service discovery port</td>
</tr>
<tr>
<td>tabadminagent.filetransfer.port</td>
<td>tabadminagent:filetransfer</td>
<td>TSM Agent file transfer port.</td>
</tr>
<tr>
<td>vizportal_0.authentication.port</td>
<td>vizportal:authentication</td>
<td>Application server authentication port.</td>
</tr>
<tr>
<td>vizportal_0.authorization.port</td>
<td>vizportal:authorization</td>
<td>Application server authorization port.</td>
</tr>
<tr>
<td>vizportal_0.maintenance.port</td>
<td>vizportal:.maintenance</td>
<td>Application server maintenance port.</td>
</tr>
<tr>
<td>Port names: syntax for json file (configKeys)</td>
<td>Port names: syntax for tsm CLI</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------------------------</td>
<td>---------------------------------</td>
<td>-------------</td>
</tr>
<tr>
<td>vizportal_0.microservice.extensions.port</td>
<td>vizportal:.microservice.extensions</td>
<td>Application server extensions port.</td>
</tr>
<tr>
<td>vizportal_0.monolith_grpc.port</td>
<td>vizporta:monolith_grpc</td>
<td>Application server GRPC port.</td>
</tr>
<tr>
<td>vizportal_0.publishing.port</td>
<td>vizportal:publishing</td>
<td>Application server publishing port.</td>
</tr>
<tr>
<td>vizportal_0.recommendations.port</td>
<td>vizportal:recommendations</td>
<td>Application server recommendations port.</td>
</tr>
<tr>
<td>vizportal_0.port</td>
<td>vizportal</td>
<td>Application server primary port.</td>
</tr>
<tr>
<td>vizportal_0.debug.port</td>
<td>vizportal:debug</td>
<td>Application server debug port.</td>
</tr>
<tr>
<td>vizportal_0.jmx.port</td>
<td>vizportal:jmx</td>
<td>Application server jmx port.</td>
</tr>
<tr>
<td>vizportal_0.jmx.rmi.port</td>
<td>vizportal:jmx.rmi</td>
<td>Application server jmx rmi port.</td>
</tr>
<tr>
<td>vizqlserver_0.port</td>
<td>vizqlserver</td>
<td>VizQL server primary port.</td>
</tr>
<tr>
<td>vizqlserver_0.debug.port</td>
<td>vizqlserver:debug</td>
<td>VizQL server</td>
</tr>
</tbody>
</table>
### Port names: syntax for json file (configKeys) | Port names: syntax for tsm CLI | Description
--- | --- | ---
 | | debug port.

| vizqlserver_0.jmx.port | vizqlserver:jmx | VizQL server jmx port.
| vizqlserver_0.jmx.rmi.port | vizqlserver:jmx.rmi | VizQL server jmx rmi port.

### Enable the JMX Ports

To help you work through a problem with Tableau Server, Tableau Support may ask you to enable the server’s JMX ports. These ports can be useful for monitoring and troubleshooting, usually with a tool like JConsole.

To enable the JMX ports on Tableau Server:

1. **Stop the server.**

2. Enter the following command:

   ```bash
   tsm configuration set -k service.jmx_enabled -v true
   ```

3. **Apply pending changes:**

   ```bash
   tsm pending-changes apply
   ```

   The `pending-changes apply` command displays a prompt to let you know this will restart Tableau Server if the server is running. The prompt displays even if the server is stopped, but in that case there is no restart. You can suppress the prompt using the `--ignore-prompt` option, but this does not change the restart behavior.

   For more information, see `tsm pending-changes apply`.

4. **Start the server.**
Important
Enabling JMX ports can introduce some security risk. To mitigate this risk, it is important to limit access to the JMX ports to the fewest number of clients that's practical for your scenario. You typically limit access using the host's firewall rules, an external security device, or routing rules.

How the JMX Ports Are Determined

By default, the JMX ports assigned dynamically, from within a range of available ports. For details on how port assignment is done, and how to override dynamic mapping, see Tableau Services Manager Ports.

Help Output for initialize-tsm Script

The following help content is the output when you run the following command:

```
sudo ./initialize-tsm -h
```

The initialize-tsm script is installed to `/opt/tableau/tableau_server-/packages/scripts.<version_code>/`

Output

REQUIRED

```
--accepteula
```
Indicate that you have accepted the End User License Agreement (EULA).
You can find the EULA in `/opt/tableau/tableau_server-/packages/docs.<version_code>`

OPTIONAL

```
-c config-name
```
Set the service configuration name.
If not set, the default is "tabsvc".

```
-d data-directory
directory
```
Set a custom location for the data directory
if it's not already set. If not set, the default is 
"/var/opt/tableau/tableau_server".

-\texttt{b bootstrap-file} Optional. Location of the bootstrap file 
downloaded from the Tableau Services Manager 
on existing node. Must be provided to join existing Tableau 
Server cluster.

-\texttt{u username} Name of the user with admin privileges 
on existing Tableau Services Manager. 
Required if -\texttt{b} option specified.

-\texttt{p password} Password for the Tableau Services Man-
ager admin user.

-\texttt{f} Bypass warning messages.

-\texttt{g} Do NOT add the current user to the 
"tsmadmin" administrative 
group, used for default access to Tableau Services Manager 
or 
to the "tableau" group, used for easier access to log files.

-\texttt{a username} The provided username will be used as 
the user to be added 
to the appropriate groups, instead of the user running the 
script. Providing both -\texttt{a} and -\texttt{g} is not allowed.

-\texttt{q} Quiet, suppress output except for errors and 
warnings.

-\texttt{i coordinationservice-client-port} Client port for the 
coordination service
Tableau Server on Linux Administrator Guide

- `e coordinationservice-peer-port` Peer port for the coordination service

- `m coordinationservice-leader-port` Leader port for the coordination service

- `t licenseservice-vendord daemon-port` Vendor daemon port for the licensing service

- `n agent-filetransfer-port` Filetransfer port for the agent service

- `o controller-port` Https port for the controller service

- `l port-range-min` Lower port bound for automatic selection

- `r port-range-max` Upper port bound for automatic selection

- `--disable-port-remapping` Disable automatic port selection

- `--unprivileged-user=<value>` Name of the unprivileged account to run Tableau Server. Default: "tableau".

- `--tsm-authorized-group=<value>` Name of the group(s) that allows authorization to access Tableau Services Manager. Default: "tsmadmin".

- `--disable-account-creation` Do not create groups or user accounts for Server and TSM authorization.
However, the values in: unprivileged-user and tsm-authorized-group will still be used in TSM configuration.

--debug
Print each command as it is run for debugging purposes. Produces extensive output.

--http_proxy=<value>
Http forward proxy for Tableau Server. Its value should be http://<proxy_address>:<proxy_port>/
For example, --http_proxy=http://1.2.3.4:3128/

--https_proxy=<value>
Https forward proxy for Tableau Server. Its value should be http://<proxy_address>:<proxy_port>/
For example, --https_proxy=http://1.2.3.4:3128/

Take care to use http_proxy environmental variable.
Do not specify the https protocol for the value of the https_proxy environmental variable.

--no_proxy=<value>
Environment variable that directs certain URLs to bypass the forward proxy. For example,

---no_proxy-y=localhost,127.0.0.1,localaddress,.localdomain.com

Related topics

- Controlling port remapping with initialize-tsm
- Install and Initialize TSM
View Server Version

The version of Tableau Server you are running is important. It determines the functionality and capabilities you have access to. Version is also important when you are upgrading, because in some cases how you upgrade depends on which version you are upgrading from, and which version you are upgrading to. Knowing your version is easy, once you understand how to find it.

- Viewing version in Tableau Server—All server users can view the version of Tableau Server from the Help menu in the server web UI.

- Viewing version in Tableau Services Manager (TSM)—TSM administrators can view the versions of TSM and Tableau Server from the TSM command line (CLI).

Viewing the server version from the Tableau Server web UI

- While logged into Tableau Server, click the information icon (❓) and About Tableau Server.
The version of Tableau Server is listed in the About Tableau Server dialog box:

```
About Tableau Server

Tableau Server Version: 2018.2.5 (2018.18.1214.0751) 64-bit Linux
©2018 Tableau Software, Incorporated and its licensors. All rights reserved.
Close
```

Viewing the server version and TSM version from the TSM command line

1. Open a command prompt as administrator on the initial node (the node where TSM is installed).

2. Run the following command:

   `tsm version`

The output displays the versions of Tableau Services Manager (TSM) and Tableau Server.

For example:

```
C:\> tsm version
Tableau Services Manager command line version
2018.18.1214.0751.
Tableau Server version 2018.18.1214.0751.
```

Short version, long version, and version_code

In most cases, when you need to know your version number, you need to know the "short" version. This version number displays in the About Tableau Server dialog box and is made up of three parts: major, minor, and maintenance versions. The short version number has this format: `nnnn.n.n`. For example: `2018.2.5`. 
A longer version includes the major and minor version numbers, as well as other build-specific information, and has this format: `nnnnn.nn.nnnn.nnnn`, for example: 20182.18.1214.0751. When you see a reference to `<version_code>` in this documentation, it means this longer version. The most common reference to this `version_code` or long version in the documentation is when we are discussing install locations or paths to the scripts or bin directories.

Tableau Server is installed in a `/packages` directory, with subdirectories that include the long version in the path name. This means that when you are navigating to a location within the `/packages` directory, you need to include the long version. For example, the `/scripts` directory: `/opt/tableau/tableau_server-\packages/scripts.<version_code>/`

For example: `/opt/tableau/tableau_server-\packages/scripts.20182.18.1214.0751`

Finding the long version number

This longer version also displays in the About Tableau Server dialog box, in parentheses:

```
About Tableau Server

Tableau Server Version: 2018.2.5 (20182.18.1214.0751) 64-bit Linux

©2018 Tableau Software, Incorporated and its licensors. All rights reserved.

Close
```

Configure Connection to External Services

Tableau supports a set of functions that your users can use to pass expressions to external services for integration with R and Python. This topic describes how to configure Tableau Server to connect to these external services. For more information about user scenarios and connecting to external services with Tableau Desktop, see Pass Expressions to External Services, in the Tableau Desktop and Web Authoring Help.
Because Tableau Server provides an authentication mechanism, it can be more secure to expose external service functionality to users through Tableau Server than in Tableau Desktop.

To allow users to publish workbooks that rely on an external service connection, you must configure Tableau Server:

1. Use the `tsm security vizql-extsvc-ssl enable` command to configure the baseline connection to RServer or TabPy. In addition to the baseline configuration parameters required by the command, you must also set the `--script-disabled` option to `true`.
2. Run the following commands:

   ````
   tsm security vizql-extsvc-ssl enable true
   tsm pending-changes apply
   ````

   The `pending-changes apply` command displays a prompt to let you know this will restart Tableau Server if the server is running. The prompt displays even if the server is stopped, but in that case there is no restart. You can suppress the prompt using the `--ignore-prompt` option, but this does not change the restart behavior. For more information, see `tsm pending-changes apply`.

**Script errors**

Tableau cannot verify that workbooks that use an external service will render properly on Tableau Server. There might be scenarios where a required statistical library is available on a user’s computer but not on the external service instance that Tableau Server is using.

For views that cannot be rendered in Tableau Server because of a script error, users will see a warning error when you publish the workbook:

> This worksheet contains external service scripts, which cannot be viewed on the target platform until the administrator configures an external service connection.
Changing the Identity Store

Infrastructure or business changes may require you to change the identity store on Tableau Server. There are two kinds of identity stores: local and external. When you installed Tableau Server you configured either a local identity store or an external identity store.

When you configure Tableau Server with a local identity store, all user and group information is stored and managed in the Tableau Server repository. In the local identity store scenario, there is no external source for users and groups.

When you configure Tableau Server with an external store, all user and group information is stored and managed by an external directory service. Tableau Server must synchronize with the external identity store so that local copies of the users and groups exist in the Tableau Server repository, but the external identity store is the master source for all user and group data. Examples of external identity stores are OpenLDAP and Active Directory.

For more information about the Tableau identity store, see Identity Store.

You can change from local store to an external store, or you can change from an external store to a local store. In either case, to change the identity store type, you complete these steps:

1. Uninstall (including manual deletion of directories) and then reinstall of Tableau Server. The procedure for full uninstall and clean install are at the end of this topic.

2. Restore content and permissions.

   In these steps the term "restore" does not refer to using the TSM maintenance restore command to restore the backup you are making. You cannot restore a backup (.tsbak) created on a Tableau Server instance that uses a different identity store than the target Tableau Server. The backup is a best practice safeguard, in case you need to go back to your original Tableau Server configuration.
Warning

Changing the installation type on Tableau Server can be a complicated and time-consuming process. To avoid data loss or orphaning of content or users, you'll need to plan this process carefully. In all cases, user filters applied to workbooks and data sources will need to be updated manually after the change.

Most importantly, determine how you will transition content and permissions to the new identity store after you reinstall Tableau Server.

Methods for restoring content and permissions

The following list describes two methods for restoring content and permissions after you reinstall Tableau Server. Select the method that best fits with your environmental requirements.

- **Method 1: Use site export and import**—In this method, you start by exporting each site in your existing deployment. Then, you install the new server and configure it for the new identity store type. You then create new users in the default site on the new server. Finally, you import all the original sites. During the import stage, you can map the original identities to the new users that you created in the default site.

  Because this method exports all content and permissions at each site, it is the best method for organizations that require a high fidelity replica of the content and permissions after the identity store change is complete. Some organizations require an identity store change as the result of an authentication change. In these cases, a different user name syntax is a often a requirement in the new model. This method, which includes a process of mapping original user names to new names, provides flexibility for such scenarios.

- **Method 2: Fresh installation; users republish content**—In this method, you install a new version of Tableau Server and select the new identity store type during
You also create new sites. You then create users and give them access, and they republish their workbooks and data sources. Unlike the other method, in this one, you do not reuse any of your existing Tableau Server infrastructure.

This method is most appropriate for smaller deployments with fairly autonomous and data savvy users. From an administrative perspective, this method is the simplest, since you’re not actively porting over content. However, because you rely entirely on users to republish content, this method may not be successful for large organizations or for those where centralized oversight of content is required.

User filters

User filters are domain-specific. Therefore, when the domain of Tableau Server changes or authentication type changes, filters no longer function as expected. Although the user filters are generated by Tableau Server, after they are set by the user, the filters are stored in the workbooks and data sources. Neither of these methods for changing the identity store modifies the contents of the workbooks or data sources.

As you plan the identity store change, you must also include a final task to correct user filtering in all workbooks and data sources with Tableau Desktop.

User names and the Tableau Identity store

If you are using Method 1, it’s helpful to understand how Tableau Server stores user names in the Tableau identity store. Tableau stores all user identities in the repository, which coordinates content permissions and site membership with various services in Tableau Server. Generally, an identity store configured for Active Directory store user names in the format, domain\username. Some organizations use a UPN (jsmith@domain.lan).

On the other hand, organizations that configure Tableau Server with local identity store usually create standard, truncated user names, such as jsmith.

In all cases, these user names are literal strings that must be unique in the Tableau identity store. If you are changing from one identity store type to another, then your target authentication, SSO, or user provisioning solution may require a specific user name format.
Therefore, to maintain all permissions, content, and user viability, one of the following must be true after you change the identity store type:

- The new user names must match the original user names, or
- The original user names must be updated to match a new format.

If an authentication change is driving the identity store change, then the target authentication scheme will likely impose a user name syntax that is different than your original user names. Method 1 includes a process where you can map original user names to new user names.

It's possible that the original user name format will work with the new authentication type. For example, if you used UPN names in a local identity store deployment, you might be able to use the same user names in an Active Directory deployment. You could also use the `domain\username` format for local identity store, as long as users continue to use that format to sign in to Tableau Server.

If you are changing from local identity store to an external Active Directory store, review the topic, User Management in Active Directory Deployments, as part of your planning process.

**Method 1: Use site export and import**

1. Export all sites on your server. See Export or Import a Site.
2. Back up, remove, and then reinstall.
3. Create new users on Tableau Server. You should have a new user that corresponds to each user on the original server.
4. Import the sites that you exported in Step 1. See Export or Import a Site. During import, you will be prompted to map the new users to the original users.

**Method 2: Fresh installation—users republish content**

Even if you do not plan to port content as part of your identity store change, we recommend that you back up the server.

1. Back up, remove, and then reinstall.
2. Create users, sites, and groups.
Inform your users of the new Tableau Server, provide them with credentials, and allow them to republish their content.

Back up, remove, and then reinstall

Both methods include the following steps:

1. Back up Tableau Server
2. Remove Tableau Server.
3. Reinstall Tableau Server with the new identity store type.

Step 1: Back up Tableau Server

As a best practice, you should back up the server before proceeding.

Follow the procedure, Create a backup using the TSM command line interface (CLI). Run the `backup` command with the `-d` option. The `-d` option adds the datestamp.

When you are finished, copy the backup file (.tsbak) to a safe location that is not a part of your Tableau Server installation.

Step 2: Remove Tableau Server

You must completely remove Tableau Server from the computer. See Remove Tableau Server from Your Computer.

Step 3: Reinstall Tableau Server with new authentication type

1. Go to the Tableau Customer Portal, sign in with your Tableau user name and password, and then download Tableau Server.
2. Install Tableau Server. See Install and Configure Tableau Server more information. During installation, you will select the new identity store type. See Configure Initial Node Settings.

LDAP Configuration Reference

This topic provides a description of all LDAP-related configuration options. The option name that you specify is dependent on the tool that you use to configure LDAP:
- **configEntities**: Options are set with a JSON file as described in `identityStore` Entity. Values that you enter as `configEntities` are validated before they are saved.

- **tsm CLI**: Options are set with the tsm command line tool as described in `tsm user-identity-store`. Values that you enter with tsm CLI are validated before they are saved.

- **configKey**: Options are set by running `tsm` configuration set Options. Alternatively, they may be included in a JSON configuration file as described in Configuration File Example. When you set an option with a `configKey`, the value that you enter is copied straight to the underlying `.yml` configuration file. Tableau Server does not validate the value. For this reason, we recommend using `configKeys` only when no option exists to set the configuration with `configEntities`, `tsm CLI`, or the TSM Web UI.

If you are configuring Tableau Server to use Active Directory, we recommend using the TSM Web UI for installation. The TSM Web UI is optimized to configure Tableau Server for Active Directory with the minimum necessary input. See Configure Initial Node Settings.

Consider using the **Tableau Identity Store Configuration Tool** to generate your LDAP json configuration file. The Tableau Identity Store Configuration Tool will also generate a list of key/value pairs that you can set by running `tsm` configuration set Options. The tool itself is not supported by Tableau. However, using a JSON file created by the tool instead of creating a file manually does not change the supported status of your server.

<table>
<thead>
<tr>
<th>configEntities (Options are case sensitive)</th>
<th>tsm CLI</th>
<th>configKey</th>
<th>Scenario</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>type</td>
<td>N/A</td>
<td>wgserv-er.authenticate</td>
<td>AD, LD, LD-</td>
<td>Where you want to store user identity information. Values: local or activedirectory.</td>
</tr>
</tbody>
</table>
If you want to connect to any LDAP server, enter `activedirectory`.

<table>
<thead>
<tr>
<th>Option</th>
<th>Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>sslPort</strong></td>
<td>N/A</td>
<td>Use this option to specify the secure port of the LDAP server. We recommend secure LDAP for simple bind. LDAPS is usually port 636.</td>
</tr>
<tr>
<td><strong>port</strong></td>
<td>N/A</td>
<td>Use this option to specify the non-secure port of the LDAP server. Plaintext is usually 389.</td>
</tr>
<tr>
<td><strong>domain</strong></td>
<td>domain</td>
<td>In Windows Active Directory environments, specify the domain where Tableau Server is installed, for example, &quot;example.lan&quot;. In LDAP directories, specify the root domain name in the same format. For example, if your root is &quot;dc=my,dc=root&quot;, specify &quot;my.root&quot;. If your root does not use a <code>dc</code> component, see the root <code>configEntity</code> option below.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>username</th>
<th>ldap-username</th>
<th>user.domain.username</th>
<th>A-D, L-DAP</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>wgserv-user-domain.username</td>
<td>The user name that you want to use to connect to the directory service.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>A-D, L-DAP</td>
<td>The account that you specify must have permission to query the directory service.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>For Active Directory, enter the user-name, for example, jsmith.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>For LDAP servers, enter the distinguished name (DN) of the user that you want to use to connect. For example, &quot;cn=jsmith,dc=example,dc=lan&quot;.</td>
</tr>
<tr>
<td>password</td>
<td>ldap-password</td>
<td>wgserv-user-domain.password</td>
<td>A-D, L-DAP</td>
</tr>
<tr>
<td>password</td>
<td></td>
<td>A-D, L-DAP</td>
<td>The password of the user account that you will use to connect to the LDAP server.</td>
</tr>
<tr>
<td>directoryServiceType</td>
<td>N/A</td>
<td>user-domain.directoryservice</td>
<td>A-D, L-DAP</td>
</tr>
<tr>
<td>directoryServiceType</td>
<td></td>
<td>A-D, L-DAP</td>
<td>The type of LDAP directory service that you want to connect to.</td>
</tr>
<tr>
<td>directoryServiceType</td>
<td></td>
<td></td>
<td>Values: activedirectory or openldap.</td>
</tr>
<tr>
<td>ker-ber</td>
<td>ker-bpriben</td>
<td>user-domain</td>
<td>A-D, L-DAP</td>
</tr>
<tr>
<td>ker-ber</td>
<td></td>
<td></td>
<td>The service principal name for Tableau Server on the host machine. The</td>
</tr>
<tr>
<td>osPrincipal</td>
<td>principal</td>
<td>domain.ldap.-principal</td>
<td>LDAP</td>
</tr>
<tr>
<td>-------------</td>
<td>----------</td>
<td>------------------------</td>
<td>------</td>
</tr>
<tr>
<td>hostname</td>
<td>hostname</td>
<td>wgserv.-domain.ldap.hostname</td>
<td>LDAP</td>
</tr>
</tbody>
</table>
| membersRetrievalPageSize | N/A | wgserv.-domain.ldap.members.retrieval.page.size | LDAP | This option determines the maximum number of results returned by an LDAP query.

For example, consider a scenario where Tableau Server is importing an LDAP group that contains 50,000 users. Attempting to import such a large number of users in a single operation is not a best practice. When this option is set to 1500, Tableau Server imports the first 1500 users in the first response. After those users are processed, Tableau Server requests the...
We recommend that you modify this option only to accommodate the requirements of your LDAP server.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>domain.ldap.connectionpool.enabled</code></td>
<td>When this option is set to <code>true</code>, Tableau Server will attempt to reuse the same connection when sending queries to the LDAP server. This behavior decreases the overhead of having to re-authenticate with the LDAP server on each new request. Connection pooling only works with simple bind and TSL/SSL bind connections. Connection pooling is not supported for GSSAPI bind connections.</td>
</tr>
<tr>
<td><code>kerberosConfig</code></td>
<td>The path to the Kerberos configuration file on the local computer. If you are installing into Active Directory, we don’t recommend using the existing Kerberos configuration file or keytab file that may already be on the domain-joined computer. See Identity Store tsm CLI: Uses tsm user-identity-store set-connection [options] command.</td>
</tr>
<tr>
<td><code>kerberosKeytab</code></td>
<td>The path to the Kerberos keytab file on the local computer. It is recommended that you create a keytab file with keys specifically for Tableau Server service</td>
</tr>
</tbody>
</table>
Tableau Server on Linux Administrator Guide

and that you do not share the keytab file with other applications on the computer. For example, on Linux, you might place the keytab file in the 
/var/opt/tableau/keytab directory.


<table>
<thead>
<tr>
<th>nickname</th>
<th>N/A</th>
<th>wgserv-er.domain.nickname</th>
<th>A-D</th>
<th>The nickname of the domain. This is also referred to as the NetBIOS name in Windows/Active Directory environments. The nickname option is required for all LDAP entities. If your organization does not require a nickname/NetBIOS, then pass a blank key, for example: &quot;&quot;.</th>
</tr>
</thead>
<tbody>
<tr>
<td>root</td>
<td>N/A</td>
<td>wgserv-er.domain.ldap.root</td>
<td>L-D-A-P</td>
<td>If you do not use a dc component in the LDAP root or you want to specify a more complex root you need to set the LDAP root. Use the &quot;o=my,u=root&quot; format. For example, for the domain, example.lan, the root would be &quot;o=example,u=lan&quot;.</td>
</tr>
<tr>
<td>server-SideSorting</td>
<td>N/A</td>
<td>wgserv-er.-domain.ldap.server_side_sorting</td>
<td>L-D-A-P</td>
<td>Whether the LDAP server is configured for server-side sorting of query results. If your LDAP server supports server-side sorting, set this option to true. If you are unsure whether your LDAP server supports this, enter false, as misconfiguration may</td>
</tr>
<tr>
<td>rangeRetrieval</td>
<td>N/A</td>
<td>Wgserver.-domain.ldap.range_retrieval</td>
<td>L-D-A-P</td>
<td>Whether the LDAP server is configured to return a range of query results for a request. This means that groups with many users will be requested in small sets instead of all at once. LDAP servers that support range retrieval will perform better for large queries. If your LDAP server supports range retrieval, set this option to true. If you are unsure whether your LDAP server supports range retrieval, enter false, as misconfiguration may cause errors.</td>
</tr>
<tr>
<td>bind</td>
<td>N/A</td>
<td>Wgserver.domain.ldap.bind</td>
<td>L-D-A-P</td>
<td>The way that you want to secure communication to the directory service. Enter simple for LDAP unless you are connecting to an LDAP server with Kerberos. For Kerberos, enter gss-api.</td>
</tr>
<tr>
<td>N/A</td>
<td>N/A</td>
<td>Wgserver.-domain.ldap.domain_custom_ports</td>
<td>L-D-A-P</td>
<td><strong>Note</strong>: This key is only supported for Tableau Server on Linux. Allows you to map child domains and their LDAP ports. Domain and port are separated by a colon (:) and each domain:port pair is separated by a comma (,) using this format: FQDN1:port,FQDN2:port</td>
</tr>
<tr>
<td><strong>Example:</strong></td>
<td>tsm configuration set -k wgserv-er.domain.ldap.domain_custom_ports -v child-domain1.lan:3269,child-domain2.lan:3269,childdomain3.lan:389</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-------------</td>
<td>----------------------------------------------------------------------------------------------------------------------------------</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>distinguishedNameAttribute</strong></td>
<td>N/A</td>
<td>wgser-ver.-domain.ldap.domain.ldap.dn Attribute L-D-A-P</td>
<td>The attribute that stores the distinguished names of users. This attribute is optional, but it greatly improves the performance of LDAP queries.</td>
<td></td>
</tr>
<tr>
<td><strong>groupBaseDn</strong></td>
<td>N/A</td>
<td>wgser-ver.-domain.ldap.group.baseDn L-D-Dn</td>
<td>Use this option to specify an alternative root for groups. For example, if all of your group are stored in the base organization called &quot;groups,&quot; then enter &quot;o=groups&quot;.</td>
<td></td>
</tr>
<tr>
<td><strong>classnames</strong></td>
<td>N/A</td>
<td>wgser-ver.-domain.ldap.group.classnames L-D-Dn</td>
<td>By default Tableau Server looks for LDAP group object classes containing the string “group”. If your LDAP group objects do not fit the default class name, override the default by setting this value. You can provide multipleclassnames separated by commas. If your group names include commas, you must escape them with a backslash (). For example, if you have a group name, groupOfNames, top, then enter &quot;groupOfNames, top&quot;.</td>
<td></td>
</tr>
<tr>
<td>Group Base Filter</td>
<td>Base Filter</td>
<td>WSG Server - Domain.ldap.group.baseFilter</td>
<td>LDAP Attribute</td>
<td></td>
</tr>
<tr>
<td>-------------------</td>
<td>-------------</td>
<td>------------------------------------------</td>
<td>----------------</td>
<td></td>
</tr>
<tr>
<td>groupBas-eFilter</td>
<td>base-filter</td>
<td>wgserv-er.-domain.ldap.group.baseFilter</td>
<td>L-D-A-P</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>The filter that you want to use for groups of users of Tableau Server. You might specify an object class attribute and an organization unit attribute. For example:</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>&quot;(&amp; (objectClass=s=groupofNames) (ou=Group))&quot;</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**tsm CLI:** Uses tsm user-identity-store set-group-mappings [options] command.

<table>
<thead>
<tr>
<th>Group Name</th>
<th>Group Name</th>
<th>WSG Server - Domain.ldap.group.name</th>
<th>LDAP Attribute</th>
</tr>
</thead>
<tbody>
<tr>
<td>groupNam-e</td>
<td>groupnam-e</td>
<td>wgserv-er.-domain.ldap.group.name</td>
<td>L-D-A-P</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The attribute that corresponds to group names on your LDAP server.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>tsm CLI:</strong> Uses tsm user-identity-store set-group-mappings [options] command.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Group Email</th>
<th>Group Email</th>
<th>WSG Server - Domain.ldap.group.email</th>
<th>LDAP Attribute</th>
</tr>
</thead>
<tbody>
<tr>
<td>groupEmai-l</td>
<td>groupemail</td>
<td>wgserv-er.-domain.ldap.group.email</td>
<td>L-D-A-P</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The attribute that corresponds to group email addresses on your LDAP server.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>tsm CLI:</strong> Uses tsm user-identity-store set-group-mappings [options] command.</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Group Description</th>
<th>Description</th>
<th>WSG Server - Domain.ldap.group.description</th>
<th>LDAP Attribute</th>
</tr>
</thead>
<tbody>
<tr>
<td>groupDescrip-tion</td>
<td>descrip-tion</td>
<td>wgserv-er.-domain.ldap.group.description</td>
<td>L-D-A-P</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The attribute that corresponds to group descriptions on your LDAP server.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>tsm CLI:</strong> Uses tsm user-identity-store set-group-mappings [options] command.</td>
<td></td>
</tr>
<tr>
<td>Option</td>
<td>Description</td>
<td></td>
<td></td>
</tr>
<tr>
<td>-----------------</td>
<td>-----------------------------------------------------------------------------</td>
<td></td>
<td></td>
</tr>
<tr>
<td>member</td>
<td>member of a group in LDAP.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>N/A</td>
<td>Specify the LDAP attribute that contains a list of distinguished names of</td>
<td></td>
<td></td>
</tr>
<tr>
<td>user-BaseDn</td>
<td>Specify the name of the LDAP attribute that stores the LDAP query for</td>
<td></td>
<td></td>
</tr>
<tr>
<td>N/A</td>
<td>Use this option to specify an alternative root for users. For example, if</td>
<td></td>
<td></td>
</tr>
<tr>
<td>classes</td>
<td>By default Tableau Server looks for LDAP user object classes containing</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

| N/A             | Specifies a list of distinguished names of users.                           |
| N/A             | Specifies the LDAP attribute that contains a list of distinguished names of |
| N/A             | Specifies the name of the LDAP attribute that stores the LDAP query for     |
| N/A             | Use this option to specify an alternative root for users. For example, if   |
| N/A             | Specifies a list of distinguished names of users.                           |
| N/A             | Specifies the LDAP attribute that contains a list of distinguished names of |
| N/A             | Specifies the name of the LDAP attribute that stores the LDAP query for     |
| N/A             | Use this option to specify an alternative root for users. For example, if   |
| N/A             | Specifies a list of distinguished names of users.                           |

Tableau Server on Linux Administrator Guide
If your names include commas, you must escape them with a backslash (\). For example, if you have a name, Names, top, then enter "Names, top".


<table>
<thead>
<tr>
<th>Field</th>
<th>Description</th>
<th>LDAP Attribute</th>
</tr>
</thead>
<tbody>
<tr>
<td>user-BaseFilter</td>
<td>base-filter</td>
<td>L-D-A-P</td>
</tr>
<tr>
<td>wgserv-user-domain.ldap.user.baseFilter</td>
<td>The filter that you want to use for users of Tableau Server. You might specify an object class attribute and an organization unit attribute. For example: &quot;(&amp; (objectClass=s/inetOrgPerson) (ou=People))&quot;</td>
<td></td>
</tr>
<tr>
<td>user-Username</td>
<td>ldap-user-name</td>
<td>L-D-A-P</td>
</tr>
<tr>
<td>wgserv-user-domain.ldap.user.username</td>
<td>The attribute that corresponds to user names on your LDAP server.</td>
<td></td>
</tr>
<tr>
<td>user-DisplayName</td>
<td>display-name</td>
<td>L-D-A-P</td>
</tr>
<tr>
<td>wgserv-user-domain.ldap.user.displayName</td>
<td>The attribute that corresponds to user display names on your LDAP server.</td>
<td></td>
</tr>
<tr>
<td>Attribute</td>
<td>LDAP Attribute</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------</td>
<td>----------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>userEmail</td>
<td>email</td>
<td>The attribute that corresponds to user email addresses on your LDAP server.</td>
</tr>
<tr>
<td>userCertificate</td>
<td>certificate</td>
<td>The attribute that corresponds to user certificates on your LDAP server.</td>
</tr>
<tr>
<td>thumbnail</td>
<td>thumbnail</td>
<td>The attribute that corresponds to user thumbnail images on your LDAP server.</td>
</tr>
<tr>
<td>jpegPhoto</td>
<td>jpegPhoto</td>
<td>The attribute that corresponds to user profile images on your LDAP server.</td>
</tr>
<tr>
<td>memberOf</td>
<td>member</td>
<td>Group that the user is a member of.</td>
</tr>
<tr>
<td></td>
<td>Config Key</td>
<td>Description</td>
</tr>
<tr>
<td>-----</td>
<td>---------------------------------------------------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td></td>
<td></td>
<td>By default Tableau Server looks for LDAP group object classes containing the string “group”. If your LDAP group objects do not fit the default class name, override the default by setting this value.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>For configEntity: This option takes a list of strings, which requires passing each class in quotes, separated by a comma (no space) and within brackets. For example: [&quot;basegroup&quot;, &quot;other-group&quot;].</td>
</tr>
<tr>
<td></td>
<td></td>
<td>For configKey: Enter each class, separated by a comma (no space) and within double quotes. For example: &quot;basegroup,othergroup&quot;.</td>
</tr>
<tr>
<td>groupClassNames</td>
<td>N/A wgserv-er.-domain.ldap.group.classnames</td>
<td>By default Tableau Server looks for LDAP user object classes containing the string “user” and “inetOrgPerson”. If your LDAP user objects do not use these default class names, override the default by setting this value.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>For configEntity: This option takes a list of strings, which requires passing each class in quotes, separated by a comma</td>
</tr>
<tr>
<td>userClassNames</td>
<td>N/A wgserv-er.-domain.ldap.user.classnames</td>
<td>By default Tableau Server looks for LDAP user object classes containing the string “user” and “inetOrgPerson”. If your LDAP user objects do not use these default class names, override the default by setting this value.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>For configEntity: This option takes a list of strings, which requires passing each class in quotes, separated by a comma</td>
</tr>
</tbody>
</table>
Calculated configKeys

The following Kerberos-related configKeys are calculated and set according to multiple environmental inputs. As such, they must be set by CLI or configEntries. Do not attempt to set these configKeys manually.

<table>
<thead>
<tr>
<th>Calculated configKey</th>
<th>To use the TSM CLI:</th>
<th>To use configEntity json:</th>
</tr>
</thead>
<tbody>
<tr>
<td>wgserver.domain.ldap.kerberos.conf, cfs.ldap.kerberos.conf</td>
<td>Set the Kerberos configuration file location with the <code>kerbconfig</code> option of tsm user-identity-store set-connection [options] command.</td>
<td>Set the Kerberos configuration file location with the <code>kerberosConfig</code> configEntity option.</td>
</tr>
<tr>
<td>wgserver.domain.ldap.kerberos.keytab, cfs.ldap.kerberos.keytab</td>
<td>Set the Kerberos keytab file location with the <code>kerb-keytab</code> option of tsm user-identity-store set-connection [options] command.</td>
<td>Set the Kerberos keytab file location with the <code>kerberosKeytab</code> configEntity option.</td>
</tr>
</tbody>
</table>
Unsupported configKeys

Some unsupported configKeys are present in underlying YAML configuration files. The following keys are not intended for standard deployments. Do not configure these keys:

- wgserver.domain.ldap.kerberos.login
- wgserver.domain.ldap.guid
- wgserver.domain.fqdn

About Tableau Server Management Add-on

Tableau Server Management Add-on is a separately licensed Add-on to Tableau Server which will provide enhanced security, manageability, and scalability capabilities for Tableau Server.

Server Management Add-on Licensing Requirements

Server Management Add-on is licensed on a per Deployment basis, which may be User-Based or Core-Based. A Deployment includes a licensed production Tableau Server installation and two licensed non-production Tableau Server installations that support the production installation. For more information on Deployment, see the EULA Documentation.

- The Server Management Add-on can only be activated on a licensed Tableau Server Deployment. This means that your Tableau Server must be first activated with a valid key that is either User-Based or Core-Based, before applying the Server Management Add-on product key. For more information on how to purchase Server Management Add-on and get the product key, contact your account manager.

- When the product key is active and enabled, you can use all the features that are included in the Add-on.

- When the Server Management Add-on product key is removed or deactivated, you will no longer be able to use the features that require a valid Server Management Add-on license. Any associated data will not be deleted. Each feature might have
slight differences on what happens when the license expires. For more information on the individual features use the links in the table below.

The following table lists the features that are included and require a valid Server Management Add-on license:

**Note:** The Resource Monitoring Tool is currently not supported on Linux installations.

<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
<th>Requirements to use the feature</th>
</tr>
</thead>
</table>
| About Tableau Content Migration   | The Content Migration Tool provides an easy way to copy or migrate content between Tableau Server projects, sites and deployments. You can do this between projects on separate Tableau Server installations (for instance, between a development instance of Tableau Server and a production instance of Tableau Server), or between projects on a single Tableau Server installation. | • Both the source Tableau Server (Server that you are moving the content from) and the target Tableau Server (Server that you are moving the content to) must have a valid Server Management Add-on license.  
• The Content Migration Tool must be installed on a version of Microsoft Windows that supports .NET 4.6.1 (Windows 7 or later, Windows Server 2008R2 or |
<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
<th>Requirements to use the feature</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tableau Server External Repository</td>
<td>Allows you to deploy Tableau Server Repository to Amazon Relational Database Service (RDS). The Tableau Server Repository is a PostgreSQL database that stores data about all user interactions, extract refreshes, and more.</td>
<td>The Tableau Server that is using an external repository must have a Server Management Add-on license.</td>
</tr>
<tr>
<td>Workload Management through Node Roles</td>
<td>Using node roles, you can configure where certain types of workloads are processed on your Tableau Server installation. The node roles features allows you to dedicate and scale resources to specific workloads (ex: extract refreshes, subscriptions).</td>
<td>The Tableau Server must have a valid Server Management Add-on license.</td>
</tr>
<tr>
<td>Key Management System</td>
<td>Gives you additional functionality to configure Tableau Server to use AWS as the KMS for extract encryption.</td>
<td>Tableau Server must have a valid Server Management Add-on license.</td>
</tr>
</tbody>
</table>

### Activating the Server Management Add-on license

The Server Management Add-on license is applied to a Tableau Server Installation and can be used for both User-Based and Core-Based installations.

Here is a quick overview of how you can activate the Server Management Add-on license on your Tableau Server Installation.

### Use the TSM web interface

1. Open TSM in a browser:

   https://<tsm-computer-name>:8850
2. Click **Licensing** on the **Configuration** tab. Click **Activate License**.

3. Enter or paste your Server Management Add-on product key and click **Activate**.

4. On the **Register** page, enter your registration information and click **Register**.

5. Follow the prompts and restart Tableau Server after registration is complete.

**Note:** If this is a new Tableau Server installation, and you apply the Server Management Add-on Key before you apply the Tableau Server product key, you will see an error. You may however, continue the installation and apply the Tableau Server product key using the same steps described above.

### Use the TSM CLI

1. Open a command prompt as administrator on a node in the Tableau Server cluster.

2. Run the following command with your Server Management Add-on product key to activate the license:

   ```bash
tsm licenses activate -k <server-management-add-on-product key>
   ```

   **Note:** If this is a new Tableau Server installation, run the command twice, first with the Tableau Server product key and then with the Server Management Add-on product key.

3. Apply the changes and restart the Server:

   ```bash
tsm pending-changes apply
   ```
About Tableau Content Migration Tool

This set of articles guides you through setting up, using, and maintaining the Tableau Content Migration Tool.

What is Content Migration Tool?

The Content Migration Tool provides an easy way to copy or migrate content between Tableau Server projects. You can do this between projects on separate Tableau Server installations (for instance, between a development instance of Tableau Server and a product installation), or between projects on a single Tableau Server installation. The Content Migration Tool User Interface walks you through the steps necessary to build a "migration plan" that you can use a single time, or as a template for multiple migrations.

Before migrating content, we recommend reviewing the Content Governance section in Tableau Blueprint.

Help and Support

If you have problems that you cannot solve with this documentation, contact Tableau Technical Support.

Getting Started with Tableau Content Migration Tool

This article will help you get started with the Tableau Content Management Tool. It contains links to other articles about information you need to prepare before installing the Content Management Tool, as well as steps to design a migration plan and upgrade existing installations.

Pre-Installation

Installation Requirements

The Content Migration Tool can only be installed on Windows operating systems. Before installing, you must be able to connect to the Tableau Server computer with the source site (the site you are migrating) and the computer with the destination site (the site you are
migrating to). Both the source and destination sites must have a valid Server Management Add-on license. For more information, see Install Tableau Content Migration Tool.

Product Compatibility with Tableau Server

- The 2019.4.x versions of Content Migration Tool support Tableau Server 2019.3.x and 2019.4.x versions.
- Content Migration Tool supports the 8 most recent versions of Tableau workbooks and data sources. The 2019.4.x version of Content Migration Tool will support migrating content saved in versions 10.5.x to 2019.4.x.

Post-Installation

Migration Plan

The Content Migration Tool walks you through migrating content across projects on a single site, to a new site on the same Tableau Server instance, and to sites that exist on different Tableau Server instances. The plan you create can be saved and used again for future migrations. For more information, see Migration Plan Overview.

Upgrading Content Migration Tool

For instructions on how to upgrade Content Migration Tool, see Upgrade Tableau Content Migration Tool.

Install Tableau Content Migration Tool

Installing Tableau Content Migration Tool is straightforward and easy.

**Note:** If you are upgrading from a legacy version of InterWorks Power Tools: Deployment (any version earlier than 2019.3), see Upgrade from Legacy Power Tools: Deployment to Tableau Content Migration Tool.

Content Migration Tool Installation Requirements

To use the Content Migration Tool tool:
The computer you install Content Migration Tool on must:

- Be running a version of Microsoft Windows that supports .NET 4.6.1 (Windows 7 or later, Windows Server 2008R2 or later).
- Be able to connect to the Tableau Server computer with the source site (the site you are migrating) and the computer with the destination site (the site you are migrating to). Both the source and destination sites must have a valid Server Management Add-on license.
- Have enough disk space on the drive where the \temp folder resides to hold a copy of all content being migrated in a single migration. This temporary content is deleted when the migration is complete.
- Have enough free disk space to hold the application and its logs.
- In addition, confirm that the REST API is enabled on Tableau Server (this is the default). Use the `tsm configuration get -k api.server.enabled` command to confirm this. A return value of `true` means the REST API is enabled. To enable the REST API, use the `tsm configuration set` command. For more information, see `api.server.enabled`.

Installing Content Migration Tool

To install Content Migration Tool:

1. Run the Content Migration Tool Setup program.

2. After reading the EULA, select I agree to the license terms and conditions, and click Install.

3. If the User Account Control dialog opens, click Yes to allow the installer to make changes.

Upgrade Tableau Content Migration Tool

Upgrading Tableau Content Migration Tool is straightforward and easy.

**Note:** If you are upgrading from a legacy version of InterWorks Power Tools: Deployment (any version earlier than 2019.3), see Upgrade from Legacy Power Tools: Deployment to Tableau Content Migration Tool.
Before Upgrading

- Running the Content Migration Tool Setup program will overwrite the previous version.
- Content Migration Tool does not support side-by-side installation of previous versions.

Upgrading Content Migration Tool

To upgrade Content Migration Tool:

1. Run the Content Migration Tool Setup program. You can download the latest version of the installer from the Tableau Releases page.

2. After reading the EULA, select I agree to the license terms and conditions, and click Install.

3. If the User Account Control dialog opens, click Yes to allow the installer to make changes.

Using Tableau Content Migration Tool

The following steps are designed to guide you through using the Tableau Content Migration Tool:

- Migration Plan Overview
  - Migration Plans: Servers
  - Migration Plans: Source Projects
  - Migration Plans: Workbooks
  - Migration Plans: Published Data Sources
  - Migration Plans: Migration Scripts
  - Migration Plans: Plan Options
- Migration Rollback
- Using the Tableau Content Migration Tool Console Runner
Migration Plan Overview

Tableau Content Migration Tool creates a streamlined process for Tableau Server migrations. The easy-to-follow plan can be audited, is repeatable, and works via a batch process so any number of workbooks and data sources can be migrated in a simple and efficient process.

The Content Migration Tool will display contextual tips to walk you through creating or editing a migration plan, with an option to roll back a migration if you choose. Once you select the source and destination sites, a summary of your migration will be displayed at the top of the screen as follows:

Limitations when Migrating Content

Before you start, make sure you understand the limitations when migrating content. For more information, see Migration Limitations.

Encryption Keys

Each migration plan file is generated with an encryption key unique to the application that created the plan. Encryption keys can be shared if the migration plan needs to be run through an application that did not originally generate the file. When sharing encryption keys, you will need to overwrite the existing key in the application to run the migration plan. To view your encryption key, select Help > Settings.

If you will be using the Content Migration Tool Console Runner for migration plans, you must specify the encryption key using the `tabcmt-runner encryption` command before running the plan. For more information, see Using the Tableau Content Migration Tool Console Runner.
Step 1: Start

The core of the migration process is creating a plan, which you can save and re-use for future migrations or modify and update as needed. The first step is choosing whether to create a new plan, or select a previously saved plan.

To create a new plan, click **Create New Plan**. If you already created a migration plan and want to use it, click **Browse for a Plan**.

By default, all of your saved migration plans will be stored in the **Tableau Content Migration Tool Plans** folder in your My Documents folder.

All migration plans are saved with a .tcxmx extension. Recently accessed plans will be listed separately to make them easy to select:

You can select a recently accessed plan and duplicate it to modify the plan and save it as a new plan. Select the plan you want to copy and click **Duplicate**.
Step 2: Planning

The Content Migration Tool guides you through building or editing your migration plan in six steps.

Click on each step for detailed instructions:

- Migration Plans: Servers
- Migration Plans: Source Projects
- Migration Plans: Workbooks
- Migration Plans: Published Data Sources
- Migration Plans: Migration Scripts
- Migration Plans: Plan Options

Step 3: Migration

Once you have completed your plan, you are now ready to run the batch process for migration. When you reach the final step of the migration, a plan summary displays for your verification:

**Review**

<table>
<thead>
<tr>
<th>Source:</th>
<th><a href="http://admin@win-vj23chvdie:80/#/site/Accounting-sandbox">http://admin@win-vj23chvdie:80/#/site/Accounting-sandbox</a></th>
</tr>
</thead>
<tbody>
<tr>
<td>Destination:</td>
<td><a href="http://admin@win-vj23chvdie:80/#/site/Accounting">http://admin@win-vj23chvdie:80/#/site/Accounting</a></td>
</tr>
<tr>
<td>Projects:</td>
<td>Default</td>
</tr>
<tr>
<td></td>
<td>Mkt-Q3</td>
</tr>
<tr>
<td></td>
<td>Mkt-Q4</td>
</tr>
<tr>
<td>Published Data Sources:</td>
<td>All data sources</td>
</tr>
<tr>
<td>Auto Archive:</td>
<td>No</td>
</tr>
</tbody>
</table>
If you want to change any aspects of your plan, you can click on a section in the left sidebar to go directly to that phase. When you are ready, click **Run** to run your migration.

When you click **Run**, the migration tool will prompt you about any unsaved elements of your plan. By default, any unsaved elements will be saved when you click **Yes**. Remember you can always keep your previous plan without making any changes by duplicating it during the Start phase of the migration process.

Your migration plan will run and a status bar displays for the overall plan progress and for each workbook being sent to the source server.

![Confirm Plan Save](image)

When the plan finishes running, you can click four tabs at the bottom of the screen for more information.

**Published Workbooks** details the newly published workbooks and the projects where they were migrated.
Published Data Sources shows the same information as the published workbooks.

The Output tab details the migration log of your plan.

You can save this log by clicking Save Log.

The Errors and Warnings tab highlights any problems that occurred during the migration.
You can correct these and rerun your plan.

When you have completed your migration and saved your plan, click **Done** to finish.

Option: Roll back

Content Migration Tool also offers you the ability to roll back to a previous migration. For more information, see Migration Rollback.

Migration Plans: Servers

The first step when creating a migration plan in the Tableau Content Migration Tool is to sign into the source server and the destination server.

In the Servers section of the Planning stage of your enterprise migration you'll see the options for signing into the source and destination servers, including server URLs and user credentials. The sites and projects you see when creating a migration plan are governed by the permissions of the user credentials you use. You can only migrate content that the user has access to.
If you do not have valid credentials and Server URLs, contact your Tableau administrator for assistance.

Step 1: Source

This is the starting point of the migration. The first set of credentials (Server URL, Username, and Password) is for your source files. If no prefix is included in the Server URL, then the Content Migration Tool will assume http://. There is an option at the bottom to Save Password. If you save your password, then it stores the encrypted password in the saved migration plan.

After you enter in the source information, you need to choose which site to use as the source. Sites are independent silos of workbooks, data, and user lists that are created within Tableau Server to group related content for selected users.
Tableau Server on Linux Administrator Guide

**Source**
Import from Saved Connection

**Server URL**
http://tableauserver.mydomain.com

**Username**
john.smith

**Password**

[ ] Save password

**Site:**
Sign In and Select a Site

Click **Sign In and Select a Site** to connect to the server and see a list of sites the user has permissions to access.
Select the site you want to use and click **OK**. The site is listed on the page. To change the selected site, click **Change Site**.

**Site:**

Accounting-sandbox  [Change Site]

**Required Permissions/Licenses:** The user account used to sign in to the source must have an Interactor or Explorer role or higher, and the following permissions for the content you want to migrate:

- View
- Download/save as
- Optional: Admin (to select workbooks, to access a user list)
Tableau Server on Linux Administrator Guide

You also must have a Server Management Add-on license on both your source and destination Tableau Servers.

Step 2: Destination

Repeat the server sign in process for the destination server (the server you are migrating content to).

**Destination**

[Import from Saved Connection ▼]

Server URL

Username

Password

[Save password ]

**Site:**

Sign In and Select a Site

If you want to save the sign in credentials for your destination server, select **Save Password**. Finally, click **Sign In and Select a Site** at the bottom. You’ll be prompted to select a site from a list of sites the user has permissions to access on the destination server. This site is the destination site.

If you are migrating your workbooks between two sites on the same server, your sign in credentials for the source and destination servers may be very similar or identical (including the Server URL). At a minimum, they will have a different site selected.
**Required Permissions/Licenses:** The user account used to sign in to the destination must have an Interactor or Explorer role or higher, and Publishing rights for that site.

**Optional: Add or Edit Saved Connections**

At the bottom of the screen is a link to **Add or edit saved connections**. Clicking this link will open the **Manage Tableau Server Connections** window.

**Note:** If you saved a connection with a password, you will be prompted to verify your Windows credentials.

Use this to add new saved connections. Saved connections can expedite your migration plan creation by saving both source and destination server sign in details. As part of creating a saved connection, you choose a site.
After you create a saved connection, it's listed when you click **Import from Saved Connections**. A saved connection will not automatically update your sign in credentials in the saved plan sign in information. These are saved separately within the plan.

**Step 3: Continue to Next Step**

After successfully signing in to both source and destination servers and selecting the sites, you are ready to continue to the next step of your migration plan creation, Source Projects. Click **Next**.

**Migration Plans: Source Projects**

The next step in creating a migration plan in the Tableau Content Migration Tool is to select the source projects. Source projects are the projects the workbooks and published data sources will be migrated from. The project or projects you choose impact which workbooks are available to migrate in the next step of the migration plan.

**Step 1: Select Your Source Project**

There are two options when selecting source projects, **All Projects**, and **Specific Projects**:

**Source Projects**

- [ ] All Projects
- [ ] Specific Projects

Workbooks and data sources from all projects will be available for migration.
The All Projects option selects all projects from the source site you specified in the Servers step. The Specific Projects option allows you to select specific projects from the source site.

You can select each project individually or use the Select All button and then clear selections for the projects you don't want to include. If you make any changes on the source site while on this step, you can use the Refresh button to update the projects list.

Step 2: Continue to the Next Step

Once you have selected source projects, you are ready to continue to the next step of the your migration plan, Workbooks. Click Next.

Migration Plans: Workbooks

You have successfully signed into both your source server and your destination server, selected projects, and chosen sites where your workbooks reside and where you want
them migrated. The next step is to prepare your workbooks for migration.

**Note:** If your workbooks or data sources include extracts, be sure you read and understand the information in Migrating Workbooks and Data Sources that use Extracts.

**Step 1: Workbook Selection**

All of the workbooks in the source site and projects appear on the **Workbook Selection** screen.

### Workbook Selection

- Specific Workbooks
- Rule Based
- All Workbooks

Unselect All (9 of 9 selected)
If you make any changes to the workbooks in the source site while on this step, you can click **Refresh** to update the workbook listings. There are several different ways to select these workbooks.

**Specific Workbooks Selection**

There are three buttons in the **Specific** section. Any choices from the Basic section will immediately include the specifically selected workbook in the migration plan. Alternately, you can individually select specific workbooks by clicking on each one.

**Select All**

This button will select or clear selection of all the workbooks in the site. If additional workbooks are added to the site after the plan is saved, they will not be automatically added the next time the plan is used.

**Display:**

**Thumbnails**

The default view shows your workbooks in thumbnail previews to help you differentiate each of them. In this view, mousing over the thumbnail will show previews of the other worksheets and dashboards within that workbook.

**List**

The list view is a more succinct listing that also provides additional information, including Workbook Name, Project, Tableau Version, and Last Modified.

Clicking on any of the column headers will sort the workbooks appropriately. Also, mousing over any of the workbooks will also provide a floating preview of the worksheets and dashboards within that workbook. List view is particularly useful if you have a large amount of workbooks in a site.
Rule Based Selection

You can use Rule Based selection to choose workbooks based on specific criteria. Rule-based options will create workbook selection criteria to be used when the migration plan is run. Be aware that selecting "all" in any of the Rule Based options is different than the Specific Workbooks selection. A rule-based "all" selection will always include all workbooks, so any newly added workbooks are included in future migrations.

- In projects
- Tagged with
- Published by

The Rule Based radio button allows you to select workbooks by using the following options:

Workbooks in projects

This menu allows you to select workbooks from specific projects.

Workbooks tagged with

This menu allows you to select workbooks by their tags.

Workbooks published by

This menu allows you to select workbooks by their author.

With each option, you can select individually or multiple by clicking on the option next to each entry. All selected workbooks will appear in the Selection Description box.

All Workbooks Selection

The last option is to select the All Workbooks radio button, which selects all workbooks in all projects in the site.
Again, using the **All Workbooks** radio button is different than selected all of the workbooks using the **Specific Workbook** method, because it will use every workbook in the source site each time the migration plan is used in the future, as well.

![Radio buttons for workbook selection](image)

When you are satisfied with your workbook selections, click **Next**.

![Next button](image)

**Step 2: Workbook Mapping**

You can now map your selected workbooks from the source file to the destination file. Mapping allows you to rename source workbooks as they are migrated and choose different destinations. You can also add mapping to change the project, prefix, or suffix for the workbooks as well. Projects can be added to the Destination in this section as well.

If you make no changes here, then the selected workbooks will simply be migrated with the same name and into the same project as the source. If you have not defined in projects in your destination site, then they will be migrated into the Default project. To add workbook mapping click the **Add Mapping** button. The following options will appear in the mapping area.
Rename Workbook

Change the destination name or project of a single workbook.

Change Project
Change the destination project for all workbooks from a source project.

Change Prefix
Remove or replace the prefix for workbooks from one or all source projects.

Change Suffix
Remove or replace the suffix for workbooks from one or all source projects.

Rename Workbook

To rename the workbook(s) that are being migrated, this transformation will allow you to filter by project in the **Source** site and provide a drop-down list to select the desired workbook. In the **Destination** field, select which project you would like the workbook to be directed as well as entering the desired new name.
Change Project

By default, the workbooks are migrated to the same project in the destination. This mapping allows you to change the destination project for all workbooks from a source project.

Add Project

When renaming the workbook or changing the project, the **Add New** option allows you to create a destination project without having to sign in to the destination Tableau Server and create the project manually. You can create projects and nested project using the **Add New** dialog box.
Change Prefix

This allows you to remove or replace the prefix for workbooks from one or all source projects.
Change Suffix

Like the prefix mapping, you can remove or replace the suffix for workbooks from one or all source projects.

Step 3: Workbook Transformations

You can change and modify your workbooks by using the Transformation step.

Transformations modify your workbooks in a specified manner. Additional transformations can be included via plug-ins or will be added in future versions of the application. Click on the Add Transformation drop-down menu to see the selection of Transformations currently available.

**Workbook Transformations**

- **Replace Action URL**
  Replaces all or part of an action URL.

- **Set Parameter Value**
  Sets the current value of a parameter.

- **Remove Image**
  Removes an image by file name.

- **Remove Tooltip Commands**
  Removes command buttons from all tooltips.

- **Replace Image**
  Replaces all or part of an image file name.

- **Replace Text**
  Replaces text in the workbook.

- **Set Zoom Control Visibility**
  Sets the visibility mode of all zoom controls.

- **Web Page URL Replacement**
  Replaces all or part of a web page URL used on dashboards.
Selecting any of the transformations will bring up the Edit Transformation window, which will allow you to customize it to your selected workbooks. All transformations will be completed in the order that they are listed from top to bottom.

For all of the different types of transformations, there are two basic steps. The first step is to make your selection for the transformation. In this case, select the workbook(s) you want to transform. The selection area is similar to the Workbook Selection section of the Planning phase with all of the features of the Basic selection radio button: Select/Unselect All, Refresh, Thumbnail Display, and List Display. At the top of the list, you can select Select All workbooks, which is an option to automatically select all workbooks for future transformations. You can also Refresh the workbook display window to reflect any changes or updates to the source site.

The second step is to use the options tab to enter the specific selections for whichever transformation you select.
Each of the workbook transformations have different values to be entered on the options tab, and the tab will have different names, depending on the transformation you're editing:

**Action URL Replacement**

Replace part or all of an URL action inside of the workbook using this transformation. On the options tab, enter the text that should be matched and its replacement value.

Example:

**URL**: www.exampledev.com
Tableau Server on Linux Administrator Guide

Match: dev

Replacement: Prod

Result: www.exampleProd.com

Set Parameter Value

Define a new parameter. On the options tab, enter the name of the Parameter, the data type from the drop-down menu, and the value.

<table>
<thead>
<tr>
<th>Parameter Name</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Type</td>
<td>Float</td>
</tr>
<tr>
<td>Value</td>
<td>0</td>
</tr>
</tbody>
</table>

Remove Images

Remove any images (such as a watermark) in the selected workbooks by entering in the file name on the options tab. There is an additional check box to receive a warning during migration if no image is found.

<table>
<thead>
<tr>
<th>File Name</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Warn when no matching images are found in a workbook.</td>
</tr>
</tbody>
</table>

Remove Tooltip Commands

Remove all of the tooltip commands from the selected workbooks. There are no additional options to define for this transformation.
Replace Images

Replace any images in the selected workbooks. On the options tab, enter the file name of the old image and file name of the replacement image. You can use a local file path or URL to replace images.

<table>
<thead>
<tr>
<th>File Name</th>
<th>Replacement Image URL</th>
</tr>
</thead>
</table>

Example:

**File Name:** C:\file\image.png

**Replacement Image URL:** http://www.exampledev.com/replacementImage.png

Zoom Control Visibility

Set the visibility mode from the drop-down menu: **Automatic, Show on Hover**, or Hide on the options tab.

| Visibility Mode | Automatic |

Web Page URL Replacement

Replace part or all of a web page URL used on dashboards using this transformation. On the options tab, enter the text that should be matched and its replacement value.

| Match | Replacement |

1742 Version: 2019.4 Tableau Software
Step 4: Data Source Transformations

The next step in planning your workbooks for your enterprise migration are your data source transformations. It is similar in function to the Workbook Transformations step. These are for data sources that are packaged within the workbooks. Published data sources are handled in a different step in the process.

Click on the **Add Transformation** drop-down menu and the following options will appear:

### Data Source Transformations

- **Replace Table/Schema Name**
  Replaces all or part of a table or schema name.

- **Set Calculation Formula**
  Overwrites the formula for a calculated field.

- **Set Data Source Display Name**
  Change the display name on a data source

- **Set Connection Info**
  Modifies connection information for matching data sources.

- **Set Custom SQL**
  Modifies the custom SQL for matching data sources.

- **Remove Extract**
  Removes extracts from matching data sources.
Selecting any of the data source transformations will bring up the Edit Transformation window, which will allow you to customize it to your selected data sources. All transformations will be completed in the order that they are listed from top to bottom.

For all of the different types of data source transformations, there are two basic steps. The first step is to enter in the match criteria for the desired data source. Depending on which connection type you select, more fields will appear on the Match Criteria tab.

Click on the Preview Source Connections to find any connections that match the criteria entered.
The second step is to use the options tab to enter the specific selections for whichever transformation you select.
Each of the data source transformations have different values to be entered on the options tab:

Set Calculation Formula

On the options tab, you can replace the calculation for a column.

Set Connection Info

On the options tab, enter in the file path for the new data source.
Set Custom SQL

On the options tab, enter in any custom SQL in the text area desired. Be aware that custom SQL can negatively impact the performance of your workbooks if improperly used.

Remove Extract

There is no options tab for this transformation, simply enter in the Match Criteria information and the extract will be removed during migration.

In addition, on each of the transformations you can enter notes in the Comments section on the left-hand side of the Edit Transformation window.

Step 5: Publish Options

The final step in the Workbooks phase is to select publish options and create transformations for tags, extract refresh schedules, and permissions.
**Workbook Publish Options**

- **Reset Dashboard Selections**
  - This option deselects all objects on dashboards.

- **Overwrite Newer Workbooks**
  - If checked, a workbook will be migrated even if it will overwrite a workbook that has been created at the same time or more recently than the moved workbook.

- **Copy Workbook Permissions**
  - When selected, the migration tool will attempt to match source workbook permissions as closely as possible.

- **Copy Extract Refresh Schedules**
  - When selected, the migration tool will attempt to set the destination workbook extract refresh schedule(s) to schedules matching the source’s name.

**Add Option**

Click on the Add Option drop-down menu for the different types of transformations you can add:
For all of the different types of transformations, there are two basic steps. The first step is to make your selection for the transformation. In this case, select the workbook(s) you want to transform. The selection area is similar to the Workbook Selection section of the Planning phase with all of the features of the Basic selection radio button: Select/Unselect All, Refresh, Thumbnail Display, and List Display. At the top of the list, you can select Select All workbooks, which is an option to automatically select all workbooks for future transformations. You can also Refresh the workbook display window to reflect any changes or updates to the source site.
The second step is to use the options tab to enter the specific selections for whichever transformation you select. **Note:** The options tab will have different names, depending on which transformation you are editing.

**Add Tags**

This allows you to add one or more tags to the workbook. If you hover your mouse over a previously entered tag, a blue “X” will appear to allow deletion.
Remove Tags

This allows you to add one or more tags to the workbook. If you hover your mouse over a previously entered tag, a blue “X” will appear to allow deletion. You can also choose to remove the tag from the source or destination workbooks.
Apply Extract Refresh Schedules

Here you can apply destination extract refresh schedules to migrated workbooks.

**Note:** The list of schedules generated are from the destination.
Set Permissions

This transformation is to edit the permissions for the selected workbooks. Enter in a Group or User and then click Add. Adjust the permissions as desired. The four different options are to Allow the permission, Deny the permission, Inherit, or to keep the Source Value.
Set Generate Thumbnail As

This allows you to set the User or Group to be used for generating user-specific data in the workbook thumbnail after being migrated. Each option has a drop down to select the desired user or group.
Step 6: Continue to Next Step

You have completed the workbooks section of the planning phase of your migration. Click Next to continue.

You will now move on to the Migration Plans: Published Data Sources section of the planning phase.

Migration Limitations

There are certain limitations to migrations using the Tableau Content Migration Tool.
Migrating Workbooks with User Configurations

The following user configurations from the source site will not be migrated to the destination site when using the Content Migration Tool:

- Data-driven alerts
- Subscriptions
- Custom views
- Favorites
- Comments

Migrating Server-level Configurations

The following server-level configurations from the source site will not be migrated to the destination site when using the Content Migration Tool:

- Users
- Project permissions and locked state
- Site settings

Migrating Tableau Prep Flows

Tableau Prep flows published to the source site are not migrated to the destination site when using the Content Migration Tool. To run flows on a schedule and refresh the flow output using Tableau Prep Conductor, users must republish flows to the destination site with Tableau Prep Builder.

Thumbnails for Workbooks and Views

Workbooks and views that are migrated using the Content Migration Tool will retain their original thumbnails, even if the migration plan includes transformations that result in the views...
being rendered differently (for example, if data connections change). To update thumbnails, edit the workbook or view on the destination site and re-save it.

Migration Plans: Published Data Sources

The next step of creating a migration plan in the Tableau Content Migration Tool is to select, map, and add any transformations to your published data sources. The process is very similar to the Workbooks step of the planning phase, particularly the data source mapping step.

**Note:** If your workbooks or data sources include extracts, be sure you read and understand the information in Migrating Workbooks and Data Sources that use Extracts.

Step 1: Selection

Starting the Published Data Sources phase of the migration plan, you’ll select any data sources you want to include in the migration plan:

<table>
<thead>
<tr>
<th>Name</th>
<th>Project</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sheet1 (state_plates)</td>
<td>Mkt-Q3</td>
</tr>
</tbody>
</table>

The data sources will only be selected at the moment of migration. You have two methods of selection. You can use the **Specific Data Sources** radio button to click on one or more published data sources. You can use the **Refresh** button to reload the list of published data sources available.
The second option is to select the All Data Sources radio button, which true to its name, selects every data source in the selected source server project.

Step 2: Mapping

The next step is to map your source data sources to the new destination. This is similar in functionality to mapping workbooks.

**Data Source Mapping**

No changes to data source names or projects.

If you make no changes here, then the selected data sources will simply be deployed with the same name and project as the source. To add data source mapping click Add Mapping. The following options will appear in the mapping area.

<table>
<thead>
<tr>
<th>Name</th>
<th>Project</th>
<th>Destination Name</th>
<th>Destination Project</th>
</tr>
</thead>
<tbody>
<tr>
<td>Delete</td>
<td>(All Selected Data Sources)</td>
<td>(Same As Source)</td>
<td></td>
</tr>
</tbody>
</table>

The entry has the following options:

**Delete**

Clicking the Delete link will delete this mapping entry.

**Name**

Use the Name menu to select the data source you wish to map. You can select (All Selected Data Sources) to choose all of the data sources.

**Project**

The Project is the project of the associated data source names.
Destination Name

By default, the Content Migration Tool will use the same Destination Name (Same As Source), keeping the original name in the Source file, but you can type in a new name here for the destination folder.

Destination Project

If projects have been defined in your destination Site, then you can choose which project to place your migrated workbooks or you can click on the Add New option to create a new project within the destination site. You can create different project destinations for individual data sources.

<table>
<thead>
<tr>
<th>Name</th>
<th>Project</th>
<th>Destination Name</th>
<th>Destination Prc</th>
</tr>
</thead>
<tbody>
<tr>
<td>Delete</td>
<td>:d Data Sources</td>
<td>Default</td>
<td>Default</td>
</tr>
</tbody>
</table>

In most situations if multiple mapping entries exist for a single data source, a validation error will be displayed and must be fixed to continue. There is one important exception to this – a data source may match both a specific selection and a project-wide mapping entry. In this instance, the more specific entry will be used.

When you have completed all of the data source mapping necessary, click Next to continue.

Step 3: Data Source Transformations

You can modify your data sources by using the transformation option.
Click **Add Transformation** and the following options will appear:

- **Replace Table/Schema Name**
  - Replaces all or part of a table or schema name.
- **Set Calculation Formula**
  - Overwrites the formula for a calculated field.
- **Set Connection Info**
  - Modifies connection information for matching data sources.
- **Set Custom SQL**
  - Modifies the custom SQL for matching data sources.
- **Remove Extract**
  - Removes extracts from matching data sources.

Selecting any of the data source transformations opens the **Edit Transformation** window. Use this to customize your selected data sources. Transformations are executed in the order they are listed, from top to bottom.

For most data source transformations, there are two basic steps. The first step is to enter in the **Match Criteria** for the desired data source. Depending on which connection type you select, more fields will appear on the **Match Criteria** tab.
The second step depends on which transformation type you are adding. Each of the data source transformations have different values to be entered on the second tab.

You can add notes in the Comments section on the left side of the Edit Transformation window for each of the transformations you add.

**Replace Table/Schema Name**

On the Options tab, you can replace all or part of a table or schema name.

**Set Calculation Formula**

On the New Calculation Formula tab, you can replace the calculation for a column.
Set Connection Info

On the **New Connection Values** tab, there will be different fields here based on the Connection Type you selected and perhaps even no fields needed at all.

File Path

Set Custom SQL

On the **New Custom SQL** tab, enter in any custom SQL in the text area desired. You should be aware that custom SQL can negatively impact the performance of your workbooks if improperly used.

Custom SQL

Remove Extract

There is no **Options** tab for this transformation. Just type the **Match Criteria** information and the extract will be removed during migration.

Step 4: Publish Options

The final step in the Published Data Source phase is to create transformations for permissions and tags and finalize the publish options specific to the data sources.
The following checkboxes offer the following additional options:

- **Overwrite Newer Data Sources**: If checked, a data source will be deployed even if it will overwrite a data source that has been updated more recently.
- **Copy Data Source Permissions**: If checked, deployment will attempt to make source published data source permissions as closely as possible.
- **Copy Extract Refresh Schedules**: If checked, deployment will attempt to set the destination data source extract refresh schedules to schedules matching the source’s name.

Click on **Add Options** for the four different types of transformations you can add:

- **Remove Tags**: Removes one or more tags from published data sources.
- **Add Tags**: Adds one or more tags to published data sources.
- **Apply Extract Refresh Schedules**: Applies destination extract refresh schedules to migrated data sources.
- **Set Permissions**: Sets the published data source-level permissions to be used during migration.

For all of the different types of transformations, there are two basic steps. The first step is to make your selection for the transformation. In this case, select **Apply to all published data sources**, which is an option to automatically select all data sources for future transformations. You can also
Refresh the data source display window to reflect any changes or updates to the source site.

The second step is to enter the specific selections for the transformation you select.

Remove Tags

Once the data sources you would like to remove tags from are selected, enter any tags you want to remove by entering them into the field at the bottom and click Add. From this screen, you can also select to remove from the source or destination data sources. If you want to remove a previously entered tag, click on it and press the delete key.
Add Tags

After selecting the data sources desired, enter any tags you want to assign by entering them into the field at the bottom and click **Add**. If you want to remove a tag, click on it and press the delete key.
In addition, on each of the transformations you can enter notes in the Comments section on the left-hand side of the Edit Transformation window.

**Apply Extract Refresh Schedules**

This transformation allows the destination extract schedules to be selected to be added to the selected migrated data sources.
Set Permissions

The last type of transformation is to edit the permissions for the selected data sources. Enter in a Group or User and click **Add**. Adjust the permissions as desired. The four different options are to Allow the permission, Deny the permission, Inherit, or to keep the **Source Value**.
Step 5: Continue to Next Step

You have completed the data sources section of the planning phase of your migration.

Click **Next** to continue to the Migration Plans: Migration Scripts section of the planning phase.

Migration Plans: Migration Scripts

The next step of creating a migration plan in the Tableau Content Migration Tool is to create any scripts you want to run with your plan before or after migration.
Step 1: Pre-Migration

The **Run Pre Migration** section of the screen is dedicated to scripts that will run prior to migration.

![Run Pre Migration interface](image)

Each field has a help icon you can get information from by moving your cursor over it. To start with your pre-migration scripts, select **Enable**, which will then activate the fields below.

### Working Directory

This is the working directory for the script. The default directory is the same folder as the migration plan. Click on the browse button to select a different folder. The **Reset** button will restore the current folder as the plan as the working directory.

### Run

This drop down allows you to choose either to run a custom script or an executable with parameters.

### Command Executable

If you selected **Executable with Parameters** from the **Run** menu, this field will appear. This is the file path to the command executable to run prior to migration. Type it in directly or use the browse button to find the executable. This is a required field.
Command Parameters

If you selected **Executable with Parameters** from the Run drop-down menu, this field will appear. Enter in command line parameters here to use with the command executable.

Script

If you selected **Custom script** from the Run menu, enter in your pre-migration script here. It will be executed as a *.cmd file. This is a required field.

Step 2: Post-Migration

The **Run Post Migration** half of the screen is dedicated to scripts that will run after migration.

Each field has a help icon you can get information from by moving your cursor over it. To start with your post-migration scripts, select **Enable**, which will then activate the fields below.

**Working Directory**

This is the working directory for the script. The default directory is the same folder as the migration plan. Click on the browse button to select a different folder. The **Reset** button will restore the current folder as the plan as the working directory.
Run

This drop down allows you to choose either to run a custom script or an executable with parameters.

Command Executable

If you selected Executable with Parameters from the Run menu, this field will appear. This is the file path to the command executable to run prior to migration. Type it in directly or use the browse button to find the executable. This is a required field.

Command Parameters

If you selected Executable with Parameters from the Run menu, this field displays. Enter in command line parameters here to use with the command executable.

Script

If you selected Custom script from the Run menu, enter in your post-migration script here. It will be executed as a *.cmd file. This is a required field.

Step 3: Continue to Next Step

When you are ready, click Next.

Migration Plans: Plan Options

The last step of creating a migration plan in the Tableau Content Migration Tool is configuring the Options section.
Each option on this screen has a question mark that will provide contextual assistance for that particular option.

Step 1: Options

There are five options.
The Plan Name is the name of the plan as it will appear in Content Migration Tool. We recommend using a user-friendly name for your plan name.

The following checkboxes offer the following additional options:

- **Automatically Create Destination Projects that do not Exist**: By default, migrations will fail if a project does not exist on the destination server. Check this option to create those projects during migration.

- **Refresh Extracts After Migration**: If selected, any data extracts will be refreshed immediately after migration if Content Migration Tool detects that they might have been modified during migration. Click the Filter link for more options.

- **Continue Migration if Workbook or Data Source Fails**: If checked, errors migrating a workbook or data source will not cause the migration to stop. The errors will be logged and the migration will continue. Errors during version control will always stop the migration.

- **Copy Owner Settings**: This will attempt to set the workbook and data source owner of the destination server to the user with a matching username and domain of the source owner.

- **Automatically create Extract Refresh Schedules that do not Exist**: Automatically creates destination extract schedules that do not exist. If not checked, source schedules that do not exist on the destination server will not be copied.

**Exclude Extract Refreshes**

By clicking on the Filter link after the Refresh Extracts After Migration option, you will be able to choose which workbooks or published data sources will be excluded from being refreshed.
Use the back and forth arrow buttons to select which items you want excluded and then click **OK**.

**Step 2: Version Control**

These options allow you to avoid losing the existing workbooks in the destination site that might be replaced by the migrated workbooks.
Click Enable to save your previous versions. You can choose to archive workbooks and/or published data sources. You'll need to select an option from the Archive To menu which lists all of the projects in your destination server. We recommend creating a special archive project just to store your versioned workbooks.

There is a refresh button at the end of the Archive To box to update any projects that have been added on Tableau Server.

Step 3: Save Plan

Once you have completed all of your selections on the Plan Options screen, you can save your plan for future use. Click Save Plan.

Your plan will be saved to Tableau Content Migration Tool folder in the My Documents folder on your local machine.

Step 4: Continue to Next Step

When you are ready, click on Verify & Run to end the Planning phase of your migration and prepare to run your plan.
Migrating Workbooks and Data Sources that use Extracts

Tableau Server users can publish extracts which are copies, or subsets of the original data. These extracts may be embedded in a workbook or a data source. By default, when you use the Tableau Content Migration Tool to migrate a workbook or data source that contains an extract, that extract is migrated along with the workbook or data source that contains it. The Content Migration Tool gives you a couple options for controlling this behavior:

- **Switching to a Live Connection**

  You can add the **Remove Extract** transformation to your migration plan to remove the extract from your workbook or data source during migration. As always, the source workbook or data source will not be modified. The copy of the workbook or data source migrated to the destination server will have the extract removed from it. This effectively switches the data connection back to a live connection.

- **Refreshing Extracts after Migration**

  You can enable the **Refresh Extracts After Migration** option in your migration plan to have an immediate extract refresh task scheduled after the workbook or data source is migrated.

  We don’t recommend using the **Refresh Extracts After Migration** option if your migration plan also uses the **Set Connection Info** transformation to change the data connection’s to point to a different set of data (for example, a different database server or database). When you change the connection information to point to different data and use the **Refresh Extracts After Migration** option, this can unintentionally expose data in a way that is a potential security issue.
Changing data connections that use extracts

Tableau data connections are either live connections that directly query a data source, or they are extracts of a data source. Extracts are copies or subsets of the original data and can be embedded in a workbook or data source. When present, the views will query data from the extract instead of the underlying data source.

Commonly, you’ll want to modify the data source connection during the migration so that it points to a different database on the destination server than it did on the source server.

For example, if you are migrating a workbook from your staging server to your production server, you will likely want to update the data connections inside the workbook to connect to your production database. You can implement this by using the Content Migration Tool Set Connection Info transformation in your migration plan. Now you have a migration plan which copies a workbook from staging to production and updates the data connections to point to the production database.

If your workbook’s data source uses an extract then you need to do a bit more work. In the current scenario, the workbook will be migrated and the live data connection updated, but the workbook’s views will still be showing the data from the staging database because the workbook still contains same extract of the staging database which it was using when it was copied from the source (staging) server. There are a few ways to address this.

Option 1: Use Published Data Sources

You can change your workbooks so that they use published data sources instead. This way, the extract will be managed as part of the published data source and migrating updates to the workbooks that use that data source can be simplified by not having to worry about the connection to the live database or the data extract.
Option 2: Remove the Extract During Migration

You can add a Remove Extract transformation to your migration plan. This will remove the extract from your workbook, effectively switching the data source to a live connection.

Option 3: Refresh the Extract After Migration

You can use the Refresh Extracts After Migration option in your migration plan. This will migrate the extract along with workbook but will schedule an immediate extract refresh task for that workbook after the migration is complete.

This option is usually not recommended when used in combination with a Set Connection Info transformation because of potential security issues that it can introduce.

The issue is that the migrated workbook on your destination server will still show the old (source) extract data for the period of time between the completion of migration and the completion of the extract refresh task. If the extract refresh task fails, then the old/source extract data will remain until the extract is refreshed.

In a scenario like we’ve outlined above, migrating from a staging to production environment, this may be acceptable but you should be aware that the users of your workbooks may not be aware that the workbook is showing old/staging data due it being recently migrated and the extract not being refreshed yet.

In other scenarios where you may be using Set Connection Info to change data connections to point to a different set of customer or client data, this could introduce serious security issues where the workbook’s extract contains data from a different client or customer until the extract has been refreshed post-migration.

One way to mitigate this issue is to implement a 2-stage migration. This approach requires you to create two migration plans, one for each step described below and ensures the workbooks and data sources have an up-to-date extract before they are accessible.
Tableau Server on Linux Administrator Guide

- **Stage 1:** Migrate your content to a project on your destination site that only administrators have access to. This migration allows you to use the **Refresh the Extract After Migration** option along with the **Set Connection Info** transformation to update the data connection, because no unauthorized users will have an opportunity to see the old data, even if the extract refresh fails.

- **Stage 2:** After stage 1 is complete and you confirm there is a successful extract refresh, run a second migration plan to migrate the content from the stage 1 destination to the final destination where it is visible to end-users.

**Consolidate Sites**

If you need to combine the content of multiple sites into a single site (if, for example, organizational restructuring has changed how your sites should be arranged), you can use Tableau Content Migration Tool to do this.

Consolidating sites using the Content Migration Tool

You can copy all the workbooks and data sources from one site to another by following the following steps:

1. Start the Content Migration Tool.

2. Select **Create New Plan**

3. On the left, under **Source**, enter the Server URL and server administrator credentials, and select the source site (the site you will be copying the workbooks and data sources from).
4. On the right, under **Destination**, enter the server information and select the target site (the site you are copying the workbooks and data sources to).
5. Click Next and confirm the All Projects option is selected on the Source Projects page.

6. Continue to the next step by clicking Next.

7. On the Workbook Selection screen, confirm the All Workbooks option is selected and click Next.

8. If you are copying published data sources:
   
   Click Selection under Published Data Sources from the menu on the left.

   Make sure that the All Data Sources option is selected.

Note: The source server and destination server might be the same, but you can also migrate content from a site on one server to a site on another server.
9. Click **Run Migration** from the menu on the left to review the migration plan. Once you’re satisfied, click **Run** at the bottom of the screen to run the migration.

10. Repeat these steps until you’ve consolidated all sites

For more information on setting up migration plans, see Using Tableau Content Migration Tool.

**Migration Rollback**

You can use the rollback feature of the Tableau Content Migration Tool when you have workbooks or data sources that have been archived during migration (see Migration Plans: Plan Options for more information on archiving). To initiate a rollback, click **Rollback** on the Start screen.

**Roll Back**

You will be guided through four preparatory steps before starting the rollback.

**Step 1: Server**

The first step is to sign into server.
 unlike the Planning phase of creating or editing a migration plan, here you only need to enter the connection information once, for the server on which you want to roll back the migration.

Click **Import from Saved Connection** to select a stored connection. Click **Add or edit saved connections** to create or update your sign in information. When you are ready, click **Sign In and Select a Site**.

If you used stored connection information, the default site will be automatically listed. You can change the site by clicking **Change Site**. Click **Next** when you are ready to continue.
Step 2: Migration

The next step is to choose which migration and archived workbooks and data sources you want to restore.

**Migration**

<table>
<thead>
<tr>
<th>Date</th>
<th>Summary</th>
</tr>
</thead>
<tbody>
<tr>
<td>7/16/2019 5:22 PM</td>
<td>ACCT-archive: 2 workbooks, 0 data sources</td>
</tr>
</tbody>
</table>

The historical migration are listed by date with a summary of each. The summary shows how many workbooks and published data sources were archived. Select the migration you want to roll back to and click **Next**.

Step 3: Workbooks

The next step is to select which of the archived workbooks you want to restore from the selected migration.
This screen is similar to the Workbooks step of the Planning phase. You can view your workbooks in thumbnail or list mode by toggling the different buttons on the right-hand side of the screen. You can reload the workbooks in the screen by clicking Refresh. You can individually select workbooks by clicking on each of them or Select All/Unselect All by clicking on the button.

Once you have made your selections, click Next.

Step 4: Published Data Sources

For this step, choose the archived published data sources that you want to restore from your selected migration. The selection process is identical to the data source selection screen from the migration plan process.
Step 5: Run Rollback

The final step of the rollback process is to verify that all of your selections are accurate for the workbooks and published data sources that want to restore.

**Verify**

- **Source:** http://win-vj23dhvudie/#!/site/Accounting-sandbox
- **Archive Date:** 7/16/2019 5:22:35 PM
- **Workbooks:** Regional [Project: Tableau Samples]
  Superstore [Project: Tableau Samples]
- **Published Data Sources:** No selections

When you are ready to continue, click **Roll Back** run the rollback.
A status bar indicator shows the progress of each step of the rollback. Depending on how many files you are restoring, this process may take several minutes.

**Running...**

Discovering Source Published Data Sources

When the rollback is complete, you will be alerted with a completed status bar.

**Finished**

Migration Complete

Beneath the status bar is a multi-tabbed text area with more information. This is similar to the finish screen of an actual migration. The first tab will indicate archived workbooks that were successfully restored. The second tab will list published data sources. The third tab is an output log that details the rollback. Finally, any errors or warnings will be listed in the final fourth tab.

Tableau Content Migration Tool
Version 2019.3.0
Build 2019.3.19.0712.1501+165d952

Started : 7/16/2019 6:35:19 PM
Plan : -
File : -
Migration ID : 8df883bb4-ebb5-46bd-8123-cf8d62337073

Save Log  Cancel  Done
When you are finished, click **Done** to return back to the main screen.

Done

### Using the Tableau Content Migration Tool Console Runner

Tableau Content Migration Tool includes a command line utility for running migrations, `tabcmt-runner.exe`, located in the installation folder. The default installation folder is usually: `%PROGRAMFILES%\Tableau\Tableau Content Migration Tool`.

**NOTE:** The `tabcmt-runner.exe` utility is not the same as the `tabcmt.cmd` command line utility which is used to configure the Content Migration Tool graphical application. For more information about `tabcmt.cmd`, see **Using the Tableau Content Migration Tool Command Line Interface**.

**Usage:**

- `tabcmt-runner [options] <plan_file.tcmx>`
- `tabcmt-runner license --remove`
- `tabcmt-runner license <new license key>`
- `tabcmt-runner license <license file path> [--passphrase=<license file passphrase>]`
- `tabcmt-runner encryption --reset`
- `tabcmt-runner encryption <new_key>`
- `tabcmt-runner improvement [on|off]`
- `tabcmt-runner --help`
- `tabcmt-runner --version`

**Options:**
Run Plan

Executes a migration plan immediately.

tabcmt-runner [options] <plan file>

Available options:

- `--logfile=<file name>` sets the file name to log output to
- `--https=<secure|legacy>` sets the HTTPS mode
- `--quiet` disables logging to stdout
- `--src-user=<username>` sets the username of the source Tableau Server connection
- `--src-password=<password>` sets the password of the source Tableau Server connection
- `--dest-user=<username>` sets the username of the destination Tableau Server connection
- `--dest-password=<password>` sets the password of the destination Tableau Server connection
Exit codes:

- 0 indicates that the migration was successful.
- 1 indicates that the migration was successful but warning messages were logged.
- 2 indicates that the migration failed. Specific errors will be included in the log output.

Show Plan Summary

Shows a summary of the migration plan and then exits.

tabcmt-runner --info <plan file>

help

Shows usage information for the command line utility.

tabcmt-runner --help

version

Shows the current application version information.

tabcmt-runner --version

encryption

Reset the encryption key, or specify a new one. You must specify the encryption key before using the tabcmt-runner utility, even if you already done so from the Content Migration Tool UI.

tabcmt-runner encryption <new_key> | --reset

improvement

Enables or disables collection of anonymous usage information by the application. This information is completely anonymous and is sent periodically to Tableau to help us improve Content Migration Tool.
Examples

Show whether the improvement program is enabled or disabled:

```
tagcmt-runner improvement
```

Enable or disable the improvement program:

```
tagcmt-runner improvement <on|off>
```

**license**

**Note:** This command is only applicable for legacy licenses.

Manages a legacy application license for the current user. When using a legacy key, to use the `tagcmt-runner` utility you must activate the license using this command, even if you already activated it from the Content Migration Tool UI.

Examples

Show the current license information:

```
tagcmt-runner license
```

Set/activate a serial key or offline license key:

```
tagcmt-runner license <key>
```

Remove/deactivate the current license:

```
tagcmt-runner license --remove
```

Set/activate using a license file:

```
tagcmt-runner license <file path> [--passphrase=<password>]
```
Example: Scripting Migration Plans

**Note:** This topic includes a sample script you can use as the basis for scripting a multi-plan migration that satisfies your needs and environment. This script is intended to be used as a sample only, and not to be run as-is. For detailed instructions on using the console runner, see Using the Tableau Content Migration Tool Console Runner.

Tableau Content Migration Tool command line utility for running migrations can be used to automate the running of a migration plan from an external scheduler (such as Windows Task Scheduler) or from a custom script. The console runner only runs one migration plan (stored in a .edt file) at a time. If you have a group of migration plans you want to run as a group, then you can use a custom script in combination with the Content Migration Tool console runner.

The example below is written in PowerShell and uses the console runner to execute a list of migration plans as a group.

The following example code demonstrates:

- Running multiple migration plans as a group using the console runner.
- Optionally halting deployment of the group of plans immediately when any single migration in the group fails.
- Using the console runner’s exit code to determine whether the migration failed or logged warnings.

```
# List of migration plans to execute as a group.
$planFiles = @( 'customer 1.tcmx',
         'customer 2.tcmx'
)

# True of false whether to continue with the next plan if a migration fails.
```
Tableau Server on Linux Administrator Guide

$continueOnFailure = $false

# Path to the CMT console runner executable
$runnerExe = 'C:\Program Files (x86)\Tableau\Tableau Content Migration Tool\tabcmt-runner.exe'

# Store the exit code from the previously run migration plan.
$lastResult = -1

# Loop through and run each migration plan one at a time.
$planFiles | % {
    $file = $_

    if ($lastResult -ge 2 -and -not($continueOnFailure)) {
        Write-Warning "Skipping plan because previous migration failed. \nSkipped plan: $file"
        return
    }

    Write-Verbose "Running migration plan: $file"
    & $runnerExe $file
    $lastResult = $LASTEXITCODE

    if ($lastResult -ge 2) {
        Write-Error "Migration failed. See output or log file for error details.\nPlan: $file" -ErrorAction 'Continue'
    }
    elseif ($lastResult -eq 1) {
        Write-Warning "Migration completed with warnings. See output or log file for warning details.\nPlan: $file"
    }
}
Using the Tableau Content Migration Tool Command Line Interface

The Tableau Content Migration Tool includes a command line interface, `tabcmt.cmd`, located in the installation folder. The default installation folder is usually: 

```%
%PROGRAMFILES%\Tableau\Tableau Content Migration Tool.
```

**Note:** The `tabcmt.cmd` utility is not the same as the Content Migration Tool console runner, `tabcmt-runner.exe`. The console runner is a separate command line utility used for running migrations from the command line. For information on using the Content Migration Tool console runner, see Using the Tableau Content Migration Tool Console Runner.

Here are the commands that can be used with the `tabcmt` command line:

- `migrate`
- `help`
- `license`
- `update`
- `version`

**migrate**

Opens a migration plan file to the migrate step in the GUI:

```
tabcmt migrate <plan file>
```

**help**

Shows general help about the command line interface and the available commands.

**Examples**

Show all commands available:

```
tabcmt help
```

Show help and usage information for a specific command:
license

Manages the application license for the current user.

**Note:** This command is only applicable for legacy licenses.

Examples

Show the current license information:

```
tabcmt license
```

Remove/deactivate the current license:

```
edt license remove
```

Set/activate a serial key or offline license key:

```
tabcmt license <key>
```

Set/activate using a license file:

```
tabcmt license <file path> [--passphrase=<password>]
```

update

Manages the options for application updates.

Examples

Show the current update settings:

```
tabcmt update
```

Enable or disable the automatic update notifications:

```
tabcmt update --disabled=<true|false>
```

Set the URL to detect/download updates from:
tabcmt update --url=<url>

Enable or disable showing beta updates. Set to false to only show stable release updates.

`tabcmt update --beta=<true|false>`

version

Shows the current application version information.

`tabcmt version`

**Tableau Content Migration Tool Settings**

The Tableau Content Migration Tool default settings work in most cases, but you can change these if you need to, or if you are working with Tableau Support and they ask you to make changes.

To view or update the Content Migration Tool settings:

1. Open Content Migration Tool.

2. Click **Help** and **Settings**. The **Settings** dialog opens:
3. **Diagnostics**—Click **Open Log Folder** to open the logs location. Here you can view the logs, and zip them up if you need to send them to Tableau. For more information, see Tableau Content Migration Tool Log Files.

Select **Enable Network Tracing** if you are working with Support and they ask you to include a network trace in the logs. This applies until you clear the option or restart the Content Migration Tool.
4. **Security**—The encryption key is automatically generated on installation. If you change the encryption key, any migration plans with embedded passwords that were created with the previous key cannot be opened. If you have multiple installations of Tableau Content Migration Tool and want to share migration plans, you need to make sure the encryption key used by each instance of the tool is the same.

5. **Tuning**—In almost all cases you can leave these set to the defaults. If you are working with Support, they may ask you to change these settings.

6. **Temporary Files**—Select a location for temporary files if you want to change the default. This is the location where content is copied during a migration. You may want to change this if the default location does not have enough space to temporarily hold migrated content.

7. **Networking**—Selecting **Allow Legacy HTTPS Connections** gives you the ability to connect to Tableau Server installations running with older HTTPS configurations (for example, SSL v3). This is not recommended.

**Tableau Content Migration Tool Log Files**

Tableau Content Migration Tool generates log files when you run migrations. These can be helpful for troubleshooting problems.

**Note:** For information on all the Content Migration Tool settings, see Tableau Content Migration Tool Settings.

**Content Migration Tool Log File Location**

To find the Content Migration Tool log files from within the Content Migration Tool:

1. Start Content Migration Tool.

2. Click **Help** and **Settings**:
3. In the **Settings** dialog, click **Open Log Folder**:

A window opens with the log files.

If you are working with Tableau Support and they ask you to send log files, zip the files up before you send them. For more information on sending log files to Tableau, see the Tableau Knowledge Base.

**Legacy Product Notes (Power Tools: Deployment)**

This section includes information about *legacy* **Power Tools: Deployment** releases version 1.7 through 1.38.1. For information on Tableau Content Migration Tool, see About Tableau Content Migration Tool.
During the period covered by these legacy notes, terminology and product branding changed. References to software/feature naming may no longer be accurate.

**Important:** The Tableau 2019.3 release of Tableau Content Migration Tool includes fixes for the following potential security issues in legacy versions of **Power Tools: Deployment**:

- Only one pre-deployment and one post-deployment script is allowed.
- Sensitive values are no longer present in the logs.

**Product Compatibility with Tableau Server**

Content Migration Tool version 2019.3 supports Tableau Server versions - 10.4 through 2019.3.

The Content Migration Tool requires the Server Management Add-on or a local, legacy license. The legacy license applies to customers who purchased Power Tools licensing prior to the Tableau acquisition of the products.

For more information about activating a legacy license, see Tableau Content Migration Tool Legacy License Key Activation. If you are a customer who is looking to purchase the Server Management Add-on for your Tableau Server and Content Migration Tool, see About Tableau Server Management Add-on.

**Enterprise Deployment Tool v1.3x**

**Power Tools for Tableau Deployment v1.38.1**

Version 1.38 is a maintenance release for Power Tools: Deployment application.

**Improvements:**

- This release fixes an error where copying data source tags could cause deployments to fail.

**Power Tools for Tableau Deployment v1.38**

Version 1.38 is a feature release for Power Tools: Deployment application.
This version has been rebranded for Tableau. The language here reflects this update. For example, “InterWorks Power Tools Deployment” refers to a legacy version, while “Tableau Power Tools Deployment” refers to this new, rebranded version.

This also includes important installer changes and upgrade considerations that you can find detailed below.

Upgrade Process:

1. Install Tableau Power Tools Deployment. Do not uninstall your current Interworks version. It needs to be in place for settings to be imported.
2. On first run, an automatic upgrade process runs in the background, and attempts to find the previous InterWorks Power Tools Deployment installation. If one is found, then all settings are automatically imported. If an installation is detected but no previous install directory can be found, then you will be is prompted to provide the installation directory.
3. Once Tableau Power Tools Deployment installation is complete and you have confirmed it is working as expected, uninstall the InterWorks Power Tools Deployment.

Improvements:

- Beta support for Tableau 2019.2. Minimum support is now Tableau 10.3.
- Rebranding for Tableau.
- Changed installer to reflect Tableau ownership. This means the Tableau-branded version is installed side-by-side with the legacy InterWorks product. Setup will attempt to automatically import the settings and load the settings and recent workbook sources from the legacy install. If the automatic import is not able to locate the legacy install, you will be able to manually select the legacy install directory for settings import and complete the installation.
- Fixed an issue where archiving to a permission-locked project could cause deployment errors.
- Fixed various problems related to deployments when users of the same username but different domains exist.
- Fixed an issue where “Open in Tableau Server” links would not work correctly when using Tableau Server 2019.1 and above.
- Fixed an error where copying refresh extract schedules would choose the wrong schedule on the destination server.
• Changed deployment concurrency to prevent an issue that could cause duplicate tags to be created in Tableau Server.

Power Tools for Tableau Deployment v1.37

Version 1.37 is a feature release for Power Tools: Deployment application.

Improvements:

• Beta support for Tableau 2019.1. Minimum support is now Tableau 10.2.
• Added validation to the “Replace Text” workbook transformation to enforce at least one search scope is selected.
• Improved “Set Custom SQL” data source transformation to handle data sources with multiple custom SQL queries.
• Fixed an issue where setting the schema for Snowflake connections in the “Set Connection Info” transformation would not update table-level schema values.

Power Tools for Tableau Deployment v1.36

Version 1.36 is a feature release for Power Tools: Deployment application.

Improvements:

• Added a new “Replace Text” workbook transformation to find and replace text in one or more areas of the workbook.

• This feature allows the user to specify what text they would like to replace and the scope at which the tool will find and replace with the new text. The steps to utilize this tool are:

  1. Select the workbook(s) you would like to apply this transformation to.

  2. Insert the text you would like to replace in the “Match” field and the desired new text in the “Replacement” field. You are able to select which scopes that the tool will look for the desired text and apply the change. If it is unable to find the
Added the ability to enable a warning message during the deployment if it is unable to find the text in the desired scopes.

- Added the ability to set the encryption mode for Teradata connections in the “Set Connection Info” transformation.

Power Tools for Tableau Deployment v1.35

Version 1.35 is a feature release for Power Tools: Deployment application.

Improvements:

- Tableau 2018.3 RTM Support.
- Extract refresh schedules can now be maintained during deployment, with the option to automatically create missing schedules on the destination server.
- Specific extract refresh schedules can be appended during deployment.
- Added “Integrated Authentication” as a connection authentication type for Teradata connections. This corresponds to Kerberos authentication.
- “Permissions & Tags” screens have been renamed to “Publish Options.”
- Moved workbook/data source specific options from the general “Options” screen to the relevant “Publish Options” screens.
- Improved deployment logs to provide more detail about what was deployed and what transformations were applied to those deployed items.
- Added the ability to remove tags from published data sources.

Power Tools for Tableau Deployment v1.34

Version 1.34 is a feature release for Power Tools: Deployment application.

Improvements:

- Beta support for Tableau 2018.3. Minimum support is now Tableau 10.1.
- Added explicit authentication type selection to Set Connection Info transformations. Most connection types support “Prompt” and “Embedded Password.”
- Added viewer credential authentication type selection to Set Connection Info transformations for several connection types.
Bug Fixes:

- Fixed an issue where Hadoop-like connections using viewer credentials would include an empty password, leading Tableau Server to show the connection information incorrectly.
- Fixed an error where unchecking the Select All checkbox on the workbook list would have selected workbooks remain if the workbook list had many items.
- Fixed several usability issues with the parent project drop down on the create project dialog.
- Fixed an error that could cause permissions to not copy during deployment when the option was selected.

Enterprise Deployment Tool v1.33

Version 1.33 is a feature release for the Enterprise Deployment Tool for Tableau.

Improvements:

- Support for Tableau 2018.2.
- Added support for nested projects on Tableau Server.
- Added a warning when embedded credentials are used alongside copying ownership settings, as they are incompatible due to Tableau API limitations.
- Changed Copy Owner Settings to default to false since it is incompatible with embedded data source credentials.
- Improvements to transformation plugin engine to support future features.
- Fixed a crash when saving deployment logs to a file that is locked by another process.

Enterprise Deployment Tool v1.2x

Enterprise Deployment Tool v1.24

Version 1.24 is a feature release for the Enterprise Deployment Tool for Tableau.
Improvements:

- Official support for Tableau v10.2.
- Minor updates/changes to EULA. Please review it for any changes relevant to your situation.
- Added the ability for the set connection info transformation to embed a tableau user into a tableau server workbook data source for permission impersonation.

Enterprise Deployment Tool v1.23

Version 1.23 is a feature release for the Enterprise Deployment Tool for Tableau.

Improvements:

- Beta support for Tableau v10.2.

Enterprise Deployment Tool v1.22

Version 1.22 is a feature release for the Enterprise Deployment Tool for Tableau.

Improvements:

- Added partial support for deploying workbooks and published data sources with cube connections. Some custom transformations may not function correctly with cube connection, and standard transformation support is still experimental.
- Added the ability to transform string extras for ODBC connections.

Enterprise Deployment Tool v1.21

Version 1.21 is a feature release for the Enterprise Deployment Tool for Tableau.

Improvements:

- Official support for Tableau v10.1.
- Added the ability to transform DSN for ODBC connections.

Bug Fixes:

- Fixed an error where certain kinds of tooltips would change during deployment.
Enterprise Deployment Tool v1.20

Version 1.20 is a feature release for the Enterprise Deployment Tool for Tableau.

Improvements:

- Enterprise Deployment Tool for Tableau is now Power Tools for Tableau Deployment!
- Beta support for Tableau v10.1.
- Support for Tableau 8.0 has now lapsed – the minimum supported version is Tableau 8.1.
- Workbook descriptions are now maintained during deployment.

Enterprise Deployment Tool v1.1x

Enterprise Deployment Tool v1.19

Version 1.19 is a feature release for the Enterprise Deployment Tool for Tableau.

Improvements:

- Official Tableau v10.0 support.

Enterprise Deployment Tool v1.18

Version 1.18 is a feature release for the Enterprise Deployment Tool for Tableau.

Improvements:

- Beta support for Tableau v10.0.

Bug Fixes:

- Fixed an issue with the “Preview Matching Connections” button being disabled due to validation errors in non-visible tabs.
- Fixed a display issue with workbook permission dialog.
- Minor styling/usability fixes.
Enterprise Deployment Tool v1.17

Version 1.17 is a feature release for the Enterprise Deployment Tool for Tableau.

Improvements:

- Official Tableau v9.3 support.
- Options dialog tabs now have friendlier labels.
- Added better handling of corrupt user config files.
- Several small usability improvements to the tag editor control.

Enterprise Deployment Tool v1.16

Version 1.16 is a feature release for the Enterprise Deployment Tool for Tableau.

Improvements:

- Beta support for Tableau v9.3.
- Added a new tag entry control to workbook selection and tag hook dialogs that allows entering arbitrary tags.
- Added a “Web Page URL Replacement” transformation to the default plugin.

Enterprise Deployment Tool v1.15

Version 1.15 is a feature release for the Enterprise Deployment Tool for Tableau.

Improvements:

- Official support for Tableau v9.2.
- Deployment plans from a newer version server to an older version server are now officially supported. A deployment error can still occur if a workbook or data source that is incompatible with the destination server is selected for deployment.

Enterprise Deployment Tool v1.14

Version 1.14 is a feature release for Workbook Tools for Tableau.
Improvements:

- Added a new “Set Display Name” data source transformation for workbook data sources.
- Added a command line option to activate a serial key, offline key, or license file.
- Added a warning when support is nearing or has expired. Can be hidden through user settings.
- Added a command line option to modify automatic update settings, or disable automatic updates.
- Application log files will now roll over due to file size, and changed log format.
- Workbook selection rule text no longer runs out of the description box.
- Reduced certain timeouts so that connections with an auto-detect channel will connect faster.

Bug Fixes:

- Fixed a styling issue with the trial banner.
- Fixed an error when trying to save a plan file with invalid file characters in the plan name.
- Fixed an error where setting a float parameter value would also reset its step value.
- Fixed an error where tooltip button setting would not be preserved for certain kinds of workbooks.

Enterprise Deployment Tool v1.13

Version 1.13 is a feature release for the Enterprise Deployment Tool for Tableau.

Improvements:

- Official Tableau v9.1 support.

Enterprise Deployment Tool v1.12

Version 1.12 is a feature release for the Enterprise Deployment Tool for Tableau.

Improvements:

- Added beta support for Tableau v9.1. A future release will add official support once 9.1 RTM is available.
Tableau 7 is also now considered deprecated. Features may continue to work against Tableau 7 for some time but no official support will be given.

- Major scalability/optimization work to permission features to operate well in environments where there are large numbers or users.
- Minor tooling updates.

Enterprise Deployment Tool v1.11

Version 1.11 is a feature release for the Enterprise Deployment Tool for Tableau.

Improvements:

- Users and groups can now be selected from a list when setting permissions. Arbitrary text is still supported.
- Removing extracts from data sources will now remove the packaged file if there are no further references to the file.
- Added the ability to remove workbook tags during deployment.
- Added a data source transformation for setting calculation formulas.
- Application styling and usability improvements.
- Improved background loading to reduce the number of login requests that are made.
- Improved scalability on systems with many users when REST API is available.

Bug Fixes:

- Fixed a crash when deploying archived workbooks in a non-rollback scenario.
- Fixed an error where data sources might not be available for selection in the published data source transformation dialog.
- Fixed an error where certain workbooks might lose legend titles during deployment.
- Fixed errors where certain workbooks could not be deployed due to overly-strict validation of XML.
- Fixed an error where temporary files would not be cleaned up when opening TDSX files.

Enterprise Deployment Tool v1.10

Version 1.10 is a feature release for the Enterprise Deployment Tool for Tableau.
Improvements:

- Permissions can now be preserved as closely as possible during deployment.
- Permissions can be modified/customized during deployment.
- Tags can be added during deployment.
- Data source transformations have been redesigned to allow for custom transformations, and to more closely match workbook transformations.
- Plan validation is now more robust.
- Published data sources can now archived, and can be rolled back.
- Several minor UI improvements to make adding mapping entries and creating projects more convenient.

Enterprise Deployment Tool v1.x

Enterprise Deployment Tool v1.9

Version 1.9 is a feature release for the Enterprise Deployment Tool for Tableau.

Improvements:

- Official support for Tableau 9.
- Added support for setting float parameters requiring greater than 64 bits of storage.
- Window size/position will now be saved between plan manager application sessions.
- Made some styling tweaks on the deployment status screen.

Bug Fixes:

- Fixed an issue where modifying the database on Teradata connections would not update fully qualified table names during deployment.
- Disabled use of REST API for workbook listing as it was causing performance issues.
  A future release will allow advanced per-connection settings to re-enable this.

Enterprise Deployment Tool v1.8

Version 1.8 is a feature release for the Enterprise Deployment Tool for Tableau.
Improvements:

- Added the option to limit source projects to work with. The selected projects will then act as filters for the rest of the plan creation and deployment process.
- Added the ability to modify custom SQL in data source transformations.
- Added a safer option to change tableau server data source references using inline data source transformations instead of replacement data sources.
- Added an option for fine grained extract exceptions.
- Added the ability to specify pre- and post-deployment scripts.
- Combined the settings dialogs.
- A new setting to log network communication between the application and servers is now available.

Enterprise Deployment Tool v1.7

Improvements:

- Partial support for Tableau Server 9.0 Beta. Workbook listing on Tableau Server 9.0 is more scalable, but may be significantly slower. If you experience very long loading times please let us know.
- EDT now requires at least Windows Vista or newer with .NET Framework v4.5 or newer installed. The installer will download and install .NET Framework 4.5 if needed.
- Added the ability to disable extract refreshing after deployment.
- Added the ability to delete plan files from the start screen.
- Added full support for Hadoop and Teradata connections to data source transformations.
- Added more support for SSL/Kerberos authentication to data source transformations.
- Tags will now be maintained for published data sources during deployment.
- Improvements to plan canceling responsiveness.
- Errors in workbook transformations will now cause deployment failure instead of warnings.
- Sorted data source transformation connection type list in alphabetical order instead of Tableau order.
- Added a feedback link to the application.
- Minor spacing improvements to the deployment log.
- Minor styling fixes/tweaks.
Bug Fixes:

- Many improvements/fixes in underlying SDKs.
- Removed server, port, and file from the data source transformation dialog for server connections because it was easy to create broken workbooks. A safer way to make changes to server connections will be available in a future release.
- Several fixes to validation handling.
- Several other minor bug fixes.

Legacy Compatibility Notes (Power Tools: Deployment)

This topic includes compatibility information about legacy Power Tools: Deployment releases through 1.35. For information on Tableau Content Migration Tool, see About Tableau Content Migration Tool.

During the period covered by these releases, terminology and product branding changed. References to software/feature naming may no longer be accurate.

Power Tools for Tableau, as a product suite, supports up to eight total versions of Tableau. The window for support shifts as new versions of Tableau are released, including beta releases.

**Current Support Range as of Nov 13, 2018**: Tableau 10.1-Tableau 2018.3

Support updates usually come in two waves: beta support and RTM support.

- Beta support
  - Adds support for the new version
  - Testing is performed on the most recent beta version supplied by Tableau, so will not account for additions/changes Tableau makes between the tested beta version and their official/RTM release.
  - Support for oldest version is dropped to make room for the beta version.
  - Targeted to release before the RTM version for those needing day-one support.
- RTM support
  - Tested against the official/RTM version.
  - Support for beta version is replaced with support for RTM version.
## Legacy Product Notes (Power Tools: Deployment)

<table>
<thead>
<tr>
<th>Version</th>
<th>release date</th>
<th>tableau compatibility</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.35</td>
<td>Nov 9, 2018</td>
<td>Tableau 10.1-2018.3</td>
</tr>
<tr>
<td>1.34</td>
<td>Oct 11, 2018</td>
<td>Tableau 10.1-2018.3</td>
</tr>
<tr>
<td></td>
<td></td>
<td>beta</td>
</tr>
<tr>
<td>1.33</td>
<td>Aug 15, 2018</td>
<td>Tableau 10.0-2018.2</td>
</tr>
<tr>
<td>1.32</td>
<td>Apr 26, 2018</td>
<td>Tableau 9.3-2018.1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>beta</td>
</tr>
<tr>
<td>1.31</td>
<td>Jan 1, 2018</td>
<td>Tableau 9.2-10.5</td>
</tr>
<tr>
<td>1.30</td>
<td>Dec 22, 2017</td>
<td>Tableau 9.1-10.5</td>
</tr>
<tr>
<td></td>
<td></td>
<td>beta</td>
</tr>
<tr>
<td>1.29</td>
<td>Nov 10, 2017</td>
<td>Tableau 9.1-10.4</td>
</tr>
<tr>
<td>1.28</td>
<td>Sep 6, 2017</td>
<td>Tableau 9.1-10.4</td>
</tr>
<tr>
<td></td>
<td></td>
<td>beta</td>
</tr>
<tr>
<td>1.27</td>
<td>Jul 17, 2017</td>
<td>Tableau 9.0-10.3</td>
</tr>
<tr>
<td>Version</td>
<td>release date</td>
<td>tableau compatibility</td>
</tr>
<tr>
<td>---------</td>
<td>--------------</td>
<td>-----------------------</td>
</tr>
<tr>
<td>1.26</td>
<td>Jun 1, 2017</td>
<td>Tableau 9.0-10.3 beta</td>
</tr>
<tr>
<td>1.24</td>
<td>Mar 17, 2017</td>
<td>Tableau 8.3-10.2</td>
</tr>
<tr>
<td>1.23</td>
<td>Feb 15, 2017</td>
<td>Tableau 8.3-10.2 beta</td>
</tr>
<tr>
<td>1.21</td>
<td>Dec 9, 2016</td>
<td>Tableau 8.2-10.1</td>
</tr>
<tr>
<td>1.20</td>
<td>Oct 21, 2016</td>
<td>Tableau 8.2-10.1 beta</td>
</tr>
<tr>
<td>1.19</td>
<td>Aug 18, 2016</td>
<td>Tableau 8.1-10.0</td>
</tr>
<tr>
<td>1.18</td>
<td>Jul 20, 2016</td>
<td>Tableau 8.1-10.0 beta</td>
</tr>
<tr>
<td>1.17</td>
<td>Apr 6, 2016</td>
<td>Tableau 8.0-9.3</td>
</tr>
<tr>
<td>1.16</td>
<td>Mar 16, 2016</td>
<td>Tableau 8.0-9.3 beta</td>
</tr>
<tr>
<td>1.15</td>
<td>Dec 8, 2015</td>
<td>Tableau 7.0-9.2</td>
</tr>
<tr>
<td>1.13</td>
<td>Sep 21, 2015</td>
<td>Tableau 7.0-9.1</td>
</tr>
</tbody>
</table>
Upgrade from Legacy Power Tools: Deployment to Tableau Content Migration Tool

This topic describes the process for upgrading from a legacy version of InterWorks Power Tools: Deployment (any version earlier than 2019.3) to the Tableau Content Migration Tool.

Upgrade Process

1. With your current InterWorks version of Power Tools: Deployment installed, install the Tableau Content Migration Tool. Your current InterWorks version needs to be in place so settings can be imported.

2. When you install the Content Migration Tool, an automatic upgrade process runs in the background. This upgrade process attempts to find the existing InterWorks Power Tools Deployment installation. If one is found, all settings are automatically imported. If an installation is detected but no previous install directory can be found, you will be prompted to provide the installation directory.

3. After the Tableau Content Migration Tool installation is complete and you have confirmed it is working as expected, uninstall the older InterWorks Power Tools: Deployment.

Tableau Content Migration Tool Legacy License Key Activation

The 2019.3 version of Power Tools: Deployment (now named the Tableau Content Migration Tool) requires the Server Management Add-on or a local, legacy license. The legacy license applies to customers who purchased Power Tools licensing prior to the Tableau acquisition of the products. If you are a customer who is looking to activate the Server Management Add-on license for your Tableau Server and Content Migration Tool, see About Tableau Server Management Add-on.
The instructions below provide the steps to activate your legacy license key based on the Activation Method you select.

Serial Key

1. Open Content Migration Tool.

2. Click Help and Settings. The Settings dialog opens.

3. Click Legacy License Key and leave the default Activation Method selected.

4. Enter your serial information in the Serial Key section and click OK.

![Settings dialog](image)

Offline Activation

If you are not able to activate Content Migration Tool directly, for example, if your computer is not connected to the internet, or has a firewall that restricts access outside your intranet, you can use the information below to activate Content Migration Tool offline.

An offline activation requires you to enter the Full License Key of your legacy license. The Full License Key was provided by InterWorks Support before the Tableau acquisition of PowerTools. If needed, you can request the Full License Key for an Offline Activation by contacting Tableau Technical Support.
License File

A License File allows you to export and import your License Key into another computer to activate Content Migration Tool. This can be used with command-line to automate the licensing import process. The License File was provided by InterWorks Support, but can still be requested by contacting Tableau Technical Support.
Key Management System

With the release of version 2019.3, Tableau Server includes an updated key management system (KMS).

Tableau Server local KMS

The Tableau Server local KMS uses the secret storage capability described in Manage Server Secrets to encrypt and store the master extract key. In this scenario, the Java key-store serves as the root of the key hierarchy. The Java keystore is installed with Tableau Server. Access to the master key is managed by native file system authorization mechanisms by the operating system. In the default configuration, the Tableau Server local KMS is used for encrypted extracts. The key hierarchy for local KMS and encrypted extracts is illustrated here:

AWS KMS for encryption at rest

If your organization is deploying Data Extract Encryption at Rest, then you may optionally configure Tableau Server to use AWS as the KMS for extract encryption. To enable AWS KMS, you must deploy Tableau Server in AWS EC2. In the AWS scenario, Tableau Server uses the AWS KMS customer master key (CMK) to generate an AWS data key. Tableau Server uses the AWS data key as the root master key for all encrypted extracts.
However, even when configured for AWS KMS, the native Java keystore and local KMS are still used for secure storage of secrets on Tableau Server. The AWS KMS is only used to encrypt the root master key for encrypted extracts.

Using AWS to encrypt the master root key provides better security properties by not storing the master key under the same permissions as the extracts.

The key hierarchy when Tableau Server is configured with AWS KMS

Configure AWS KMS for Tableau Server encrypted extracts

To use the AWS customer master key (CMK) to encrypt the root key in the Tableau Server KMS hierarchy, you must configure Tableau Server as described in this section.

Before you begin, verify that you meet the following requirements:

- Tableau Server must be deployed in AWS EC2
- Tableau Server must be configured with a Server Management Add-on license. See About Tableau Server Management Add-on.
- You must have administrative control of a customer master key (CMK) created in AWS Key Management Service
Step 1: Create CMK and set key policy for Tableau Server in AWS

The following procedures are performed in the AWS KMS service. References are included to AWS documentation.

1. Create the CMK that you will use for Tableau Server. See the AWS topic, Creating Keys.
2. Update the server instance’s IAM role.

Tableau Server needs to be able to authenticate with AWS KMS using the instance’s IAM role. The role needs to have a policy attached to it. The policy should be giving the instance permissions to call the "GenerateDataKey" and "Decrypt" actions on the CMK. See IAM Roles for Amazon EC2..

In a multinode deployment of Tableau Server, all nodes of the server must be running under roles that have this policy (or equivalent) attached. You can assign the same role to all nodes in the cluster.

3. At a minimum, the CMK must have a key policy where the Effect is set to Allow the Principal (the IAM role that is attached to the server instances) the Action: GenerateDataKey and Decrypt. See Using Key Policies in AWS KMS.

Step 2: Collect AWS configuration parameters

You will need the full ARN string from AWS KMS. This string is in the "General configuration" section of the AWS KMS management pages. The ARN is presented in this format: arn:aws:kms:<region>:<account>:key/<CMK_ID>, for example, arn:aws:kms:us-west-2:867530990073:key/1abc23de-fg45-6hij-7k89-1l0m-n1234567.

You will also need to specify the AWS region, which is also included in the ARN string. In the example above, the region is us-west-2. The region is where your KMS instance resides. In the next step, you will need to specify a region as shown in the Region column in the Amazon API Gateway table.

Step 3: Configure Tableau Server for AWS KMS

Run the following command on Tableau Server. This command will restart the server:
The --key-arn option takes a direct string copy from the ARN in the "General configuration" section of the AWS KMS management pages.

For example, if your AWS KMS instance is running in us-west-2 region, your account number is 867530990073, and your CMK key is 1abc23de-fg45-6hij-7k89-1l0mn1234567, then the command would be as follows:

```bash
tsm security kms set-mode aws --aws-region "us-west-2" --key-arn "arn:aws:kms:us-west-2:867530990073:key/1abc23de-fg45-6hij-7k89-1l0mn1234567"
```

Step 4: Enable encryption at rest

See Extract Encryption at Rest.

Step 5: Validate installation

1. Run the following command:

```bash
tsm security kms status
```

The following information may be returned:

- The ARN (ID) of the customer master key (CMK)
- The region the CMK is in
- The ID of the root master key (RMK) in use. The RMK is a key that is encrypted by the CMK. Tableau Server decrypts the CMK by making calls to AWS KMS. The RMK is then used to encrypt/decrypt the master extract key (MEK). The RMK can change, but there will be only one at a time.
- KMS stores a collection of master extract keys (MEKs). Each MEK has:
  - An ID, for example, 8ddd70df-be67-4dbf-9c35-1f0aa2421521
  - Either a "encrypt or decrypt key" or "decrypt-only key" status. If a key is "encrypt or decrypt", Tableau Server will encrypt new data with it. Otherwise, the key will only be used for decryption
2. View logs after you encrypt and decrypt extracts:

- Publish extracts to your site and then encrypt them. See Extract Encryption at Rest.
- Access the extracts with Tableau Desktop or with Web Authoring on a browser (this will decrypt the extracts for use).
- Search the vizqlserver_node log files for the `AwsKmsEncryptionEnvelopeAccessor` and `AwsKmsEncryptionEnvelope` strings. The default location of the logs are at `/var/opt/tableau/tableau_server/data/tabsvc/logs/

Log entry examples that indicate successful configuration include the following:

- Decrypted the RMK with ID 1abc23de-fg45-6hij-7k89-1l0mn1234567 using the CMK with ARN `arn:aws:kms:us-west-2:867530990073:key/1234567d-a6ba-451b-adf6-3179911b760f`
- Using RMK with ID 1abc23de-fg45-6hij-7k89-1l0mn1234567 to decrypt KMS store

For publishing and extract refreshes related to KMS, search the backgrounder logs. For more information about logs, see Server Log File Locations.

Troubleshoot configuration

Multi-node misconfiguration

In a multi-node setup for AWS KMS, the `tsm security kms status` command may report healthy (OK) status, even if another node in the cluster is misconfigured. The KMS status check only reports on the node where the Tableau Server Administration Controller
process is running and does not report on the other nodes in the cluster. By default the Tableau Server Administration Controller process runs on the initial node in the cluster.

Therefore, if another node is misconfigured such that Tableau Server is unable to access the AWS CMK, those nodes may report Error states for various services, which will fail to start.

If some services fail to start after you have set KMS to the AWS mode, then run the following command to revert to local mode: tsm security kms set-mode local.

Refresh AWS CMK

Refreshing the AWS CMK is a task that you perform with AWS. By default, the AWS CMK will refresh once a year. See the AWS topic, How Automatic Key Rotation Works. Since the ARN and region do not change, you do not need to update the KMS configuration on Tableau Server for normal CMK refresh scenarios.

After AWS CMK refreshes, you must regenerate the internal RMK and MEKs on Tableau Server. You should also re-encrypt all extracts with the new CMK:

1. Run the `tsm security regenerate-internal-tokens` command to regenerate all internal keys on Tableau Server, including the RMK and MEKs used for extract encryption.
2. Run `tabcmd reencryptextracts <site-name>` to re-encrypt extracts on a given site. Run this command on every site where you are storing encrypted extracts. Depending on the number of encrypted extracts on the site, this operation could consume significant server processing load. Consider running this operation outside of business hours. See Extract Encryption at Rest.

Regenerate RMK and MEK on Tableau Server

To regenerate the root master key and the master encryption keys on Tableau Server, run the `tsm security regenerate-internal-tokens` command.

Back up and restore with AWS KMS

A server backup can be taken in AWS mode with no additional configurations or procedures. The backup contains encrypted copies of the RMK and MEKs. Decrypting the keys requires
access and control of the AWS CMK.

For the restore scenario, the server being restored to can be in either KMS mode, including Local. The only requirement is that the server the backup is being restored to has decrypt access to the CMK the backup itself used.

Upon restore, the MEKs from the backup are imported as decrypt-only keys. The RMK is not migrated over. A new RMK is generated as part of the installation/restore process.

Tableau Server External Repository

Starting in 2019.3, you can deploy Tableau Server Repository to Amazon Relational Database Service (RDS). The Tableau Server Repository is a PostgreSQL database that stores data about all user interactions, extract refreshes, and more. For more information about what Tableau Server Repository is in general, see:

- Workgroups Database
- Collect Data with the Tableau Server Repository

Tableau Server can now be configured in two ways:

**Local repository:** The PostgreSQL Database is installed and deployed locally, meaning it is deployed along with Tableau Server. This is what is available all the versions of Tableau Server previous to version 2019.3.

**External repository:** The PostgreSQL Database is deployed externally. In this release, deploying the repository to Amazon RDS PostgreSQL DB instance is supported.

This topic and the information detailed in this topic is about Tableau Server external repository.

External Repository Considerations

Amazon RDS offers scalability, reliability, high availability and security built-in for PostgreSQL. By integrating with AWS to configure Tableau Server external repository, you will
be able to take advantage of these additional benefits of deploying the cloud.

Requirements

- Your Tableau Server must be installed and running on Amazon Web Services (AWS).
- Your Tableau Server must be using version 2019.3 or later.
- Your Tableau Server must have the Server Management Add-on keys activated.
- Familiarity with Amazon RDS setup and management.

Managing the External Repository

License Management

To configure an external Tableau Server Repository, you must first enable this feature using Server Management Add-on product keys. For more information, see About Tableau Server Management Add-on. If you don’t have the Server Management Add-on keys activated or if the licenses is expired, you will see the following behavior:

- If you try to configure Tableau Server to use external repository during installation, you will see an error message, but you will be able to continue the installation and Tableau Server Repository will be installed locally. For single server installations, the repository is installed on the same machine as Tableau Server. For multi-node installations, the repository is installed on one of the nodes of your Tableau Server cluster.

- If you are already using an external Tableau Server Repository on your Tableau Server installation, and the Server Management Add-on license expires, the server will fail on restart. If you no longer have a valid Server Management Add-on license, but have a valid Tableau Server license, you will still be able to create a backup. You can also migrate the external repository to local repository which does not require the Server Management Add-on license, to get your server up and running again. For more information on how to migrate from external repository to local repository, see Re-Configure Tableau Server Repository.
Supported Migration Scenarios

- Moving your repository from local to Amazon RDS
- Moving your repository from Amazon RDS to local

Backup and Restore

The backup and restore process remains the same for both local and external repository and as described in the Backup Tableau Server data topic.

- The backup and restore commands work in the same manner for both local and external repository. Backup in the case where Tableau Server uses external repository requires more disk space for backup, so you should keep that in mind when configuring your disk space.

- The default and other supported locations for storing the backup file are the same whether it the repository is local or external.

The following exceptions apply to Tableau Servers using external repository:

- Any custom user accounts that are created in the external Tableau Server repository will be included in the restore, but the passwords for the custom user accounts will not be. The passwords will have to be reconfigured after restore is complete. Custom user accounts are PostgreSQL database user accounts, used by SQL or other database client software to connect to the PostgreSQL database.

  Note: These custom accounts will be disabled as a security measure, but this can be reconfigured.

- Configuration and topology should not be included in the backup. For more information on how to export configuration and topology settings, see Perform a Full Backup and Restore of Tableau Server.
High Availability Considerations

Tableau Server does not manage or setup high availability for the external repository. Amazon RDS offers high availability features that can be used to provide high availability, manage failover, etc. For more information, see Amazon RDS High Availability.

Monitoring the Status of the Repository

TSM status page will show the Tableau Server external service as an additional node for your Tableau Server installation.

The topology tab indicates whether there are Tableau Server external services configured:
Getting Logs from the RDS Instance

Tableau Server logging and Tableau Server log file snapshot do not include logs from the external repository. For information on setting up logging for your external repository and publishing them, see PostgreSQL Database Log Files.

Next Steps

- Create a PostgreSQL DB Instance on AWS Relational Database Service (RDS)
- Install Tableau Server with External PostgreSQL Repository
- Re-Configure Tableau Server Repository
  - Migrate from local to external
  - Migrate from external to local
Create a PostgreSQL DB Instance on AWS Relational Database Service (RDS)

This topic describes how to create a PostgreSQL DB instance to use as your Tableau Server external repository.

Requirements and Recommendations

- At a minimum use db.m4.2xlarge Amazon RDS memory optimized instance type. This is the minimum recommended AWS RDS instance size to use for Tableau Server external repository, but the exact requirements will vary with your requirements and usage. We recommend db.r4.xlarge (or db.r5.xlarge) Amazon RDS memory optimized instance types for maximum performance.

  You can start with a smaller instance type and if you find later that you need a larger instance type, you can upgrade your existing RDS instance. For more information, see Upgrading your RDS Instance.

  Secure communications between Tableau Server and the external PostgreSQL DB instance using SSL is required.

- The PostgreSQL DB instance must be reachable by all nodes in the Tableau Server cluster. One of the ways you can do this is by making the PostgreSQL DB instance a member of a security group that has the necessary permissions to be accessed by all the nodes in the Tableau Server cluster.

Create a PostgreSQL DB instance on Amazon RDS

Step 1: Create a parameter group

PostgreSQL parameters that you set for a local PostgreSQL instance in the postgresql.conf file are maintained in the DB parameter group for your DB instance. When you create a DB instance, the parameters in the associated DB parameter group are loaded.
You can modify parameter values by changing values in the parameter group. You can also change parameter values, if you have the security privileges to do so, by using the ALTER DATABASE, ALTER ROLE, and SET commands.

There are two types of PostgreSQL parameters, static and dynamic. Static parameters require that the DB instance be rebooted before they are applied. Dynamic parameters can be applied immediately.

The following table shows parameters that you should modify with Tableau recommended values:

<table>
<thead>
<tr>
<th>Name</th>
<th>Data Type</th>
<th>Parameter Type</th>
<th>Recommended Tableau Configuration Values</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>autovacuum</td>
<td>boolean</td>
<td>Dynamic</td>
<td>1</td>
<td>Starts the autovacuum subprocess.</td>
</tr>
<tr>
<td>client_min_messages</td>
<td>string</td>
<td>Dynamic</td>
<td>error</td>
<td>Sets the message levels that are sent to the client.</td>
</tr>
<tr>
<td>escape_string_warning</td>
<td>boolean</td>
<td>Dynamic</td>
<td>0</td>
<td>Warns about backslash () escapes in ordinary string literals.</td>
</tr>
<tr>
<td>lc_messages</td>
<td>string</td>
<td>Dynamic</td>
<td>en_US.UTF-8</td>
<td>Sets the language in which messages are displayed.</td>
</tr>
<tr>
<td>lc_monetary</td>
<td>string</td>
<td>Dynamic</td>
<td>en_US.UTF-8</td>
<td>Sets the locale for formatting monetary amounts.</td>
</tr>
<tr>
<td>lc_numeric</td>
<td>string</td>
<td>Dynamic</td>
<td>en_US.UTF-8</td>
<td>Sets the locale for formatting numbers.</td>
</tr>
<tr>
<td>lc_time</td>
<td>string</td>
<td>Dynamic</td>
<td>en_US.UTF-8</td>
<td>Sets the locale for formatting date and time values.</td>
</tr>
<tr>
<td>Name</td>
<td>Data type</td>
<td>Parameter Type</td>
<td>Recommended Tableau Configuration Values</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>-----------</td>
<td>----------------</td>
<td>------------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>log_auto-vacuum_min_duration</td>
<td>integer</td>
<td>Dynamic</td>
<td>100</td>
<td>Sets the minimum execution time above which auto-vacuum actions will be logged.</td>
</tr>
<tr>
<td>log_connections</td>
<td>boolean</td>
<td>Dynamic</td>
<td>1</td>
<td>Logs each successful connection.</td>
</tr>
<tr>
<td>log_lock_waits</td>
<td>boolean</td>
<td>Dynamic</td>
<td>1</td>
<td>Logs long lock waits.</td>
</tr>
<tr>
<td>log_min_duration_statement</td>
<td>integer</td>
<td>Dynamic</td>
<td>100</td>
<td>Sets the minimum execution time above which statements will be logged.</td>
</tr>
<tr>
<td>log_min_messages</td>
<td>string</td>
<td>Dynamic</td>
<td>error</td>
<td>Sets the message levels that are logged.</td>
</tr>
<tr>
<td>log_temp_files</td>
<td>integer</td>
<td>Dynamic</td>
<td>128</td>
<td>Logs the use of temporary files larger than this number of kilobytes.</td>
</tr>
<tr>
<td>rds.force_ssl</td>
<td>boolean</td>
<td>1</td>
<td></td>
<td>Force SSL connections; requires import of certs on client keystore.</td>
</tr>
<tr>
<td>rds.logical_replication</td>
<td>boolean</td>
<td>1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ssl</td>
<td>boolean</td>
<td>Static</td>
<td>1</td>
<td>Set to allow SSL connections</td>
</tr>
<tr>
<td>standard_conforming_strings</td>
<td>boolean</td>
<td>Dynamic</td>
<td>0</td>
<td>Causes … strings to treat backslashes literally.</td>
</tr>
<tr>
<td>Name</td>
<td>Data type</td>
<td>Parameter Type</td>
<td>Recommended Tableau Configuration Values</td>
<td>Description</td>
</tr>
<tr>
<td>------------</td>
<td>-----------</td>
<td>----------------</td>
<td>------------------------------------------</td>
<td>-------------</td>
</tr>
<tr>
<td>work_mem</td>
<td>integer</td>
<td>Dynamic</td>
<td>16384</td>
<td>Sets the maximum memory to be used for query workspaces.</td>
</tr>
</tbody>
</table>

For more information and a full list of PostgreSQL Parameters, see [Working with PostgreSQL Parameters](#), on the AWS site.

Step 2: Create a PostgreSQL DB instance on Amazon RDS

To create a new PostgreSQL DB instance, follow the instructions provided on the [Amazon documentation site](#).

Following are configuration options and the recommended values for the new PostgreSQL DB instance:

**• Instance Specifications**

- Use the parameter group created in **Step 1**.
- Use DB instance class that is `db.m4.2xlarge` or larger.
- Allocate at least 100GB of storage.
- Storage type and Provisioned IOPS: leave default (recommendations may change depending on load testing).

**• Settings**

- You **must** use `rails` as the master username.

  This is a requirement for the external repository to work properly with Tableau Server.

- Pick a password that meets AWS's requirements.
Network and Security

- Make sure that the RDS instance can be reached by all the Tableau Server nodes. This most often involves creating a security group that allows access from the nodes.

Database Options

- Don't create an initial database. The Database name should be left blank, as Tableau Server will create the needed databases in the RDS instance.
- The port can be anything, but we recommend leaving it as the default 5432.
- Set the DB Parameter Group to the one created in Step 1.
- Leave the IAM DB authentication as disabled.

Encryption

- You can choose whether or not you want encryption.

Backup

- This is for AWS's automated backups, not Tableau Server's backups. You can specify the settings that meets the requirements.

Monitoring

- You can specify the settings based on your requirements.

Log Exports

- You can specify the settings based on your requirements.

Maintenance
• Disable auto minor version upgrade. Tableau Server is built to use a specific version of PostgreSQL. and you will be prompted to upgrade the PostgreSQL version if needed, during Tableau Server Upgrade.

• **Delete Protection**

  • You can specify the settings based on your requirements.

**Step 3: Get the PostgresSQL DB Instance Endpoint**

After creating the PostgreSQL database instance, you can't use it until it's completed initialization by AWS and this can take several minutes. Once the instance is ready, get the endpoint information that you will use to configure Tableau Server to use this instance for the Tableau Sever Repository.

**Step 4: Download the SSL certificate file (.pem file)**

SSL is required to connect to the DB instance. You will need the .pem file when you configure Tableau Server to use the external DB instance for your Tableau Server Repository. For more information, see Using SSL to Encrypt the Connection to a DB Instance.

**Configuring High Availability for your PostgreSQL DB**

Tableau Server does not manage or setup high availability for the external repository. Amazon RDS offers high availability features that can be used to provide high availability, manage failover, etc. For more information, see Amazon RDS High Availability.

**Install Tableau Server with External PostgreSQL Repository**

This topic describes how to install and configure Tableau Server to use an external service for Tableau Server Repository.

**Before you install**

• You must have the Server Management Add-on license activated on your Tableau Server to make sure your environment has the appropriate license to use external repository. For more information about Server Management Add-on, see About
Your Tableau Server environment must be deployed on AWS cloud services.

You will need the endpoint of your PostgreSQL DB instance on Amazon RDS to configure Tableau Server to use Amazon RDS PostgreSQL DB instance as the external repository. For more information on how to create and prepare a PostgreSQL DB instance see Create a PostgreSQL DB Instance on AWS Relational Database Service (RDS).

Download the SSL certificate (.pem file) for your Amazon RDS PostgreSQL DB instance. For more information on how to download the certificate file, see Using SSL to Encrypt the Connection to a DB Instance.

**Note:** Configuring Tableau Server with external repository can only be done using TSM CLI in this release for both Windows and Linux environments. The steps are described below.

### Install and Configure

**Step 1: Create a configuration file**

Create a json file with the following configuration settings:

```json
{
   "flavor":"rds",
   "masterUsername":"rails",
   "masterPassword":"<password>",
   "host": "<RDS instance host name>",
   "port": 5432
}
```

- **Flavor:** This is the type of external service you are going to use for Tableau Server repository. In this release, only Amazon RDS is supported so the value must be "rds".
- **masterUsername:** Use "rails" for the user name. This is the user that you specified
when creating the RDS instance.

You must use "rails" as the masterUsername. This is required for the external repository to work with Tableau Server properly.

- masterPassword: This is the same password you specified when creating the RDS instance.

- host: This is the endpoint of your PostgreSQL DB instance.

- Port: The database port you specified when creating the PostgreSQL DB instance.

Step 2: Install Tableau Server and Configure the External Repository

Using TSM CLI:

1. Install and Initialize TSM: Follow the instructions provided in this topic and complete steps 1-5 which runs the setup program and installs TSM.

2. Activate and Register Tableau Server: Provide the Tableau Server Key and the IT Management Add-on key in the activate step. You will need to run the following command twice, first with the Tableau Server product key and then with the Server Management Add-on product key:

   tsm licenses activate -k <product key>

3. Configure Initial Node Settings: Follow the instructions provided in the topic to configure the initial node settings.

   Important! Do not run the Initialize and Start Tableau Server step when you configure the initial node. After completing the other steps in the Configure Initial Node Settings topic, return to this page and follow the rest of the instructions.

4. Configure Tableau Server to use the external repository by using the following commands:
Specify the external repository settings using the json file that you created in the previous step:

```
tsm topology external-services repository enable -f <filename>.json -c <ssl certificate file>.pem
```

The json file is the file that you created in the first step with the configuration settings. The SSL certificate file can be downloaded as described in this topic.

Apply the changes:

```
tsm pending-changes apply
```

Step 3: Complete tsm Initialize

To initialize and start Tableau Server:

```
tsm initialize --start-server --request-timeout 1800
```

Step 4: Complete the install

Add an Administrator Account and complete the installation.

Re-Configure Tableau Server Repository

Your Tableau Server may be configured to use either a local or an external repository. This topic describes the steps needed to reconfigure your existing Tableau Server with one of the following options:

- Move a local Tableau Server Repository to Amazon RDS and configure your Tableau Server to use an external repository.

- Move the Tableau Server Repository from Amazon RDS to your local Tableau Server installation, and configure your Tableau Server to use the local repository. This means that the Tableau Server repository will be installed on the same machine or machines as your Tableau Server.
To learn more about these options and external repositories, see Tableau Server External Repository.

Move a local Tableau Server Repository to Amazon RDS

You can move the Tableau Server Repository to Amazon RDS instance. Use the following steps to move Tableau Server Repository to Amazon RDS:

1. Activate the Server Management Add-on product key on your Tableau Server if it is not already activated. Server Management Add-on license is required to configure your Tableau Server with an external repository.
2. Configure Amazon PostgreSQL DB instance to use as the external repository. For instructions on how to configure the PostgreSQL DB instance, see Create a PostgreSQL DB Instance on AWS Relational Database Service (RDS).
3. Create a json file with the following configuration settings:

   ```json
   {
     "flavor":"rds",
     "masterUsername":"rails",
     "masterPassword":"<password>",
     "host":"<RDS instance host name>",
     "port":5432
   }
   ```

   - **Flavor:** This is the type of external service you are going to use to configure for your Tableau Server repository. In this release, only Amazon RDS is supported so the value should be “rds”.

   - **masterUsername:** Use "rails" for the user name. This is the user that you specified when creating the RDS instance.
You must use "rails" as the masterUsername. This is required for the external repository to work with Tableau Server properly. This is also the user you specified when creating the RDS instance.

- masterPassword: This is the same password you specified when creating the RDS instance.
- host: This is the endpoint of your PostgreSQL DB instance.
- Port: The database port you specified when creating the PostgreSQL DB instance.

4. Run the following TSM CLI command to configure Tableau Server to use external repository:

```
  tsm topology external-services repository enable -f file.json -c <ssl certificate file>.pem
```

The json file is the file that you created in the first step with the configuration settings. The SSL certificate file can be downloaded as described in this topic.

Running the above command will migrate the local repository to your new RDS PostgreSQL DB instance.

Move the external Tableau Server Repository to the local installation of Tableau Server

You can move the Tableau Server Repository from Amazon RDS to your local installation of Tableau Server.

1. Run the following TSM CLI command to move the repository to a specific node:

```
  tsm topology external-services repository disable -n nodeN
```

2. If you are setting up HA for your repository, install the repository on a second node.
For more information, see Example: Install and Configure a Three-Node HA Cluster.
Note: To install the repository on a second node, you must run the command described in the previous step first. The first step migrates your external repository to the local repository. You can then install the repository on a second node on your Tableau Server.

Upgrading your RDS Instance

If you find that the current RDS instance you are using to host Tableau repository is a performance bottleneck, you can upgrade your RDS instance to a larger size. This topic describes the steps that you can use to upgrade your RDS instance.

1. Back up Tableau Server data.

2. Stop Tableau Server:

   tsm stop

3. After confirming that the server has shutdown, sign in to the AWS Management Console and open the Amazon RDS console at


4. In the navigation pane, choose Databases, and then choose the DB instance that you want to modify.

5. Choose Modify. The Modify DB Instance page appears.

6. Modify your RDS instance by setting the DB instance class to the one you want.

7. Choose Apply immediately to make sure that the changes are applied right away.
   
   For more detailed information, see Modifying a DB Instance Running the PostgreSQL Database Engine on AWS documentation site.

8. Monitor the status of the RDS instance in the AWS console. It may take a while, but when the status shows as Available, you can start Tableau Server and resume nor-
Workload Management through Node Roles

Using node roles, you can configure where certain types of workloads are processed on your Tableau Server installation. The node roles features allows you to dedicate and scale resources to specific workloads. You can configure node roles for Backgrounder and File Store.

Backgrounder Node Roles

The Backgrounder process runs Tableau Server tasks, including extract refreshes, subscriptions, flow tasks, ‘Run Now’ tasks, and tasks initiated from tabcmd. Running all these tasks can use a lot of machine resources. If you have more than one Backgrounder node in your cluster, you can manage your Backgrounder workload by specifying the type of tasks a Backgrounder can run on a node using the Backgrounder node role feature.

This configuration option is currently available only through TSM CLI commands and is only useful on multi-node clusters. If you have only one node, the Backgrounder is set to run all tasks by default and that cannot be changed.

Using Backgrounder Node Roles

The Backgrounder node role feature is intended to give you more control and governance over where certain type of Backgrounder workloads are processed in your Tableau Server installation and allows you to dedicate and scale resources to specific workloads.

For example, if your deployment is heavy on extract and users are running a lot of extract refreshes or encryption jobs, it could be beneficial to dedicate a node to extract refreshes. Similarly, in the case of subscriptions, if your Tableau Server installation processes a lot of subscriptions and you want to ensure that other jobs do not take resources from subscriptions, then you can dedicate a node to subscriptions. In these cases, you would also
want to dedicate other backgrounder nodes to workloads other than extract refreshes or subscriptions.

To support high availability, Tableau recommends having multiple nodes that are dedicated towards a specific workload. For example, if you dedicate a node to extract refreshes, you should also configure a second node to process extract refresh workload. This way if a node dedicated to extract refreshes becomes unavailable, extract refreshes can still be processed by the other node.

Configuration Options

<table>
<thead>
<tr>
<th>Configuration</th>
<th>Jobs</th>
</tr>
</thead>
<tbody>
<tr>
<td>all-jobs (default)</td>
<td>All Tableau Server jobs</td>
</tr>
<tr>
<td>flows</td>
<td>Flow run jobs.</td>
</tr>
<tr>
<td>no-flows</td>
<td>All jobs except flows.</td>
</tr>
<tr>
<td>extract-refreshes</td>
<td>Jobs that are created for:</td>
</tr>
<tr>
<td></td>
<td>Incremental refreshes, full refreshes, encryption and decryption of</td>
</tr>
<tr>
<td></td>
<td>all extracts including extracts that flow outputs create.</td>
</tr>
<tr>
<td>subscriptions</td>
<td>Subscription jobs</td>
</tr>
<tr>
<td>extract-refreshes-and-subscriptions</td>
<td>Extract-refreshes, encryption and decryption of all extracts including extracts that flow outputs generate, and subscription jobs.</td>
</tr>
<tr>
<td>no-extract-refreshes</td>
<td>All jobs except extract-refreshes, extract encryption and decryption of all extracts including extracts created from flow outputs.</td>
</tr>
<tr>
<td>no-subscriptions</td>
<td>All jobs except subscriptions.</td>
</tr>
<tr>
<td>no-extract-refreshes-and-</td>
<td>All jobs except extract-refreshes, encryption and decryption of all extracts including extracts created from flow outputs, and subscriptions.</td>
</tr>
<tr>
<td>subscriptions</td>
<td></td>
</tr>
</tbody>
</table>
For more information on how to use the tsm commands to set the node role, see tsm topology.

**Note:** Making configurations to node roles require a restart of the server and will require some downtime. For more information, see tsm pending-changes.

License Requirements

Configuring a node to do only a specific type of tasks, like, flows, extract refreshes, and subscriptions, you must have one of the following licenses activated on your Tableau Server:

- To configure a node to run flows, you must have a valid Data Management Add-on license activated on your server, and have Tableau Prep Conductor running on that node. To learn more about Tableau Prep Conductor, see Tableau Prep Conductor.

- To configure a node to run all jobs except flows, you must have Tableau Prep Conductor running on the node. Tableau Prep Conductor requires a valid Tableau Data Management Add-on license activated on your server. For more information, see Licensing Tableau Prep Conductor.

- To configure a node to run extract refreshes, subscriptions, and any combination related to extract refreshes and subscriptions you must have a valid Server Management Add-on license activated on your Tableau Server. If the license expires or is deactivated, you will see an error any time you make a change to the Server configuration. For more information on Server Management Add-on, see About Tableau Server Management Add-on.

**Important!**

While flows, extract refreshes, and subscriptions can be expensive and resource heavy, they are not the only jobs that may require dedicated resources. In the **all jobs** group, there are a variety of System jobs that the Backgrounder executes, such as thumbnail generation for workbooks. Make sure that the nodes that run jobs other than extract refreshes, subscriptions, or flows have enough machine resources.
For more information on configuring node roles using TSM commands, see `tsm topology set-node-role`.

Considerations

There are some rules you must consider when configuring Backgrounder node roles, which are listed below:

- Only one node role configuration can be set for a node at a time. You cannot configure multiple node roles on a node.

- To configure a node role, there must be at least one Backgrounder process on that node.

- If you have only one Backgrounder node, you must configure this node to run all jobs. This is the default configuration and does not require additional licensing.

- If you have more than one Backgrounder node, combined, they must be configured to handle all jobs. This can be achieved in the following ways:
  
  - Configure one of the nodes to run all jobs using the `all jobs` option. This is the easiest and most straightforward way.
  
  - Using one of the exception configurations on one of the nodes:
    
    - no-flows
    - no-subscriptions
    - no-extract-refreshes
    - no-extract-refreshes-and-subscriptions

For example, in a cluster where there are three backgraders, you could have one node configured to run flows, one to run subscriptions and extract refreshes, and one to run all jobs except flows, subscription and extract refreshes.

**Note:** The ability to specify node roles to run flows, or run all jobs except flows, or run all jobs was introduced in 2019.1.
File Store Node Roles

The Tableau Server File Store controls the storage of extracts. There are three broad categories of workloads that are extract dependent.

<table>
<thead>
<tr>
<th>Extract Workload</th>
<th>Execution Service</th>
</tr>
</thead>
<tbody>
<tr>
<td>Refresh</td>
<td>Backgrounder</td>
</tr>
<tr>
<td>Query</td>
<td>Data Engine</td>
</tr>
<tr>
<td>Backup/Restore</td>
<td>Backup/Restore</td>
</tr>
</tbody>
</table>

File Store node role management in combination with Backgrounder node role management gives server admins the ability to specialize server nodes to preferentially execute selected workloads to optimize performance of all categories of extract heavy workloads.

It is possible to specialize a node to execute extract query workloads through a topology that has only stand-alone Data Engine nodes. For more information, see Optimize for Query Heavy Environments. However, this is at the expense of extract refresh workloads, which are executed by Backgrounder nodes. With the topology-based isolation approach, extract refresh heavy Backgrounder workloads can get slower as none of the Backgrounder nodes have a File Store and thus all extract refresh traffic goes over the network.

With the File Store Node Role configuration option, it is possible to designate certain server nodes that process extract queries to be preferentially selected from the list of server nodes that can do so. This helps speed up workloads such as backup and extract refreshes by allowing server admins to enable File Store on Backgrounder server nodes, which prevents extract queries from running on these nodes. This feature is useful if you have an extract-heavy query workload and an extract-heavy refresh workload and want to achieve optimal extract query and refresh performance.

Here are guidelines to optimize for extract-refresh and backup/restore workloads.
Start from a topology with specialized Data Engine nodes (see Optimize for Query Heavy Environments).

<table>
<thead>
<tr>
<th>Process</th>
<th>Initial Node</th>
<th>Additional Node 1</th>
<th>Additional Node 2</th>
<th>Additional Node 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cluster Controller</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Gateway</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Application Server</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>VizQL Server</td>
<td>✓ ✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Cache Server</td>
<td>✓ ✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Search &amp; Browse</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Back grounder</td>
<td>✓ ✓ ✓</td>
<td>✓ ✓ ✓</td>
<td>✓ ✓ ✓</td>
<td>✓ ✓ ✓</td>
</tr>
<tr>
<td>Data Server</td>
<td>✓ ✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Data Engine</td>
<td>✓ ✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>File Store</td>
<td>✓ ✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Repository</td>
<td>✓ ✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

Topology 1 - Dedicated Data Engine Nodes

1. Add File Store to Node 0.
   
   `tsm topology set-process -n node0 -pr filestore -c 1`

2. Designate Node 2 and Node 3 to preferentially execute extract-query workloads
   
   `tsm topology set-node-role -n node2, node3 -r extract-queries`

3. Designate Node 0 to preferentially execute extract-refresh workloads.
   
   `tsm topology set-node-role -n node0 -r extract-refreshes`

4. Designate Node 1 to preferentially execute non-extract-refresh workloads.
   
   `tsm topology set-node-role -n node0 -r no-extract-refreshes`

5. Apply pending changes.
   
   `tsm pending-changes apply`
**Note:** In your Tableau Server deployment, adding File Store roles to existing nodes will temporarily increase network I/O between all File Store nodes while the new File Store is being synchronized. The duration of this operation is dependent on the volume of data on the File Store and the network bandwidth capacity. The status of synchronization can be monitored using the TSM Web GUI. If you are adding more than one File Store to your deployment, it is recommended to add them consecutively and wait for the initial synchronization to complete in between each File Store addition.

**Configuration Options**

<table>
<thead>
<tr>
<th>Configuration</th>
<th>Jobs</th>
</tr>
</thead>
<tbody>
<tr>
<td>all-jobs (default)</td>
<td>All Tableau Server jobs</td>
</tr>
<tr>
<td>extract-queries</td>
<td>Jobs that are created for extract queries. The nodes selected will run as all-jobs and will prioritize the processing of extract queries.</td>
</tr>
</tbody>
</table>

For more information on configuring node roles using TSM commands, see tsm topology set-node-role.
License Requirements

To configure a node to run extract queries you must have a valid Server Management Add-on license activated on your Tableau Server.

Use the Data Management Add-on

The Data Management Add-on is a collection of features and functionality that helps customers manage Tableau content and data assets in their Tableau Server or Tableau Online environment.

Starting in Tableau Server version 2019.1, Tableau Prep Conductor is available for on-premise Tableau Server deployments, and in version 2019.3, Tableau Prep Conductor is available for Tableau Online deployments. You can use Tableau Prep Conductor to leverage the scheduling and tracking functionality of Tableau Server or Tableau Online to automate updating flow outputs.

Starting in version 2019.3, Tableau Catalog is included in the Data Management Add-on, making a variety of additional features available to you in the data management space. You can use Tableau Catalog to discover data, curate data assets, communicate data quality, perform impact analysis, and trace the lineage of data used in Tableau content.

Both Tableau Catalog and Tableau Prep Conductor are licensed through the Data Management Add-on. For more information about licensing, see License the Data Management Add-on.

To purchase the Data Management Add-on, contact your account manager.

For more information about the features in the Data Management Add-on, see the following articles:

- Tableau Prep Conductor
- About Tableau Catalog
License the Data Management Add-on

For more information on how to purchase the Data Management Add-on, contact your account manager.

The Data Management Add-on can only be activated on a licensed Tableau Server Deployment. A Deployment includes a licensed production Tableau Server installation and licensed non-production Tableau Server installations that support the production installation. For more information on Deployment, see the EULA Documentation.

Tableau Prep Conductor

- When the Data Management Add-on is active and enabled, you can publish flows to Tableau Server or Tableau Online, schedule or run flows, manage and monitor flows.
- When the Data Management Add-on is removed or deactivated, or if the Data Management Add-on expires, then the ability to publish, schedule, run flows, or receive flow alerts is disabled. You will not be able to see the flows, but the data will not be deleted.
- If your Tableau Server or Tableau Online license is still active and valid, you can download the flows using the Tableau Server REST API. For more information, see Flow Methods.

Tableau Catalog

- When the Data Management Add-on is active and enabled, you can use Tableau Catalog to discover data, curate data assets, perform impact analysis, and trace the lineage of data used in Tableau content.
- When the Data Management Add-on is removed, deactivated, or expired, the information remains on the server. The Tableau Catalog-specific information is then only accessible using the Tableau Metadata API; it no longer appears in the product. For more information, see the Metadata API.
- When the Data Management Add-on is removed, deactivated, or expired, the write APIs for all new Tableau Catalog information (for example, table descriptions, data quality warnings, column descriptions) are disabled. You can still read information using the Metadata API, however permissions on tables and databases can’t be explicitly managed in the product.
How Data Management Add-on licensing works

A Tableau Server Deployment may be user-based or core-based.

User-Based

A User-Based license metric allows you to deploy Tableau Server on a single computer or on multiple computers in a cluster. Each user that accesses Tableau Server must be licensed. Administrators add users and license them. The first Creator or Explorer product key that you add to Tableau Server activates Tableau Server and will be used by a Server Administrator.

The Data Management Add-on product key enables the features that are included with the add-on at the deployment level and those features are licensed for all the users that are already licensed for Tableau Server.

Core-Based

A Core-Based license metric imposes no constraints on the number of user accounts in Tableau Server. Instead, the license specifies the maximum number of computer cores on which you can run Tableau Server.

Typically, the total number of cores in all the computers should not exceed the total number that the Tableau Server license allows. When you add the Data Management product key to your Tableau server, it includes a specific number of Tableau Prep Conductor cores. In this scenario, the total number of cores on all computers should not exceed the total number that the Tableau Server license and the Data Management license together allow.

In this topic we will refer to the cores licensed through Tableau Server license as Tableau Server cores, and the cores licensed through Data Management Add-on as Tableau Prep Conductor cores.

Here are some concepts that apply to how licensing is applied in a Core-Based metric:

- A node can be licensed by only one of type—Tableau Server Cores or Tableau Prep Conductor cores.
The Tableau Prep Conductor cores are applied to any node that is dedicated to running Tableau Prep Conductor and when the Backgrounder on that node is set to run only flow background jobs. In this case, the total number of cores on this node cannot exceed the number of cores that Data Management Add-on license allows. If this node has any other licensed process besides Tableau Prep Conductor, Backgrounder and Data Engine enabled, then this node will require and use a Tableau Server core license.

As mentioned above, the Backgrounder node role also affects which license is used by a node. For example, if the Backgrounder node role is set to run jobs of all types (this is the default), then this node will be licensed through the Tableau Server cores. For more information on node roles, see Node Roles in Tableau Server.

See the following table and decision flow to understand how a node is licensed:

<table>
<thead>
<tr>
<th>If a node has...</th>
<th>the core on the node is counted towards...</th>
<th>the node is licensed using...</th>
</tr>
</thead>
<tbody>
<tr>
<td>one of the following processes enabled:</td>
<td>Total count of Tableau Server cores.</td>
<td>Tableau Server cores.</td>
</tr>
<tr>
<td>• Application Server</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Backgrounder (node role is set to run all jobs)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• File Store</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• Data Server</td>
<td></td>
<td></td>
</tr>
<tr>
<td>• VizQL Server</td>
<td></td>
<td></td>
</tr>
<tr>
<td>only the following processes enabled:</td>
<td>Total number of Tableau Prep Cores purchased through the Data Management Add-on.</td>
<td>Tableau Prep Cores included in the Data Management Add-on.</td>
</tr>
<tr>
<td>set to run only flows)</td>
<td>Data Engine</td>
<td></td>
</tr>
<tr>
<td>-----------------------</td>
<td>-------------</td>
<td></td>
</tr>
<tr>
<td><strong>Note:</strong> If there are no Tableau Prep Conductor cores available, but Tableau Server cores are available, then Tableau Server cores are used.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Note:** If there are no Tableau Prep Conductor cores available, but Tableau Server cores are available, then Tableau Server cores are used.
For more information about licensed processes, see Tableau Server Processes.

To learn more about Tableau Server licensing, see Tableau Server Licensing Overview.
Tableau Prep Conductor

Overview

If you build flows in Tableau Prep, you can automate running your flows to refresh flow input and output data on a schedule using Tableau Server, instead of opening Tableau Prep Builder to run individual flows manually.

Tableau Prep Conductor is one of the processes on Tableau Server. It runs flows, checks connection credentials, and sends alerts if a flow fails. Tableau Prep Conductor leverages the scheduling and tracking functionality of Tableau Server so you can automate running flows to update the flow output instead of logging into Tableau Prep Builder to manually run individual flows as your data changes.

Tableau Prep Conductor is licensed through the Data Management Add-on. To use Tableau Prep Conductor, you must have the Data Management product key activated on your server. For more information, see Licensing Tableau Prep Conductor.

Tableau Prep Conductor uses the following components to run flows:

- **Backgrounder**: Tableau Prep Conductor uses the Backgrounder process to run flows. Backgrounder is single threaded, so each instance of the Backgrounder process on a node can run one flow at a time. By adding more Backgrounders to a node, you can increase the number of flows that can be run in parallel on that node. The Backgrounder processes can be up to half the number of the physical cores of that node.

- **Connectors**: Prep Conductor uses the supported Tableau Data connectors to connect to data. For a list of supported Connectors, see Supported Connectors.

- **Data Engine**: Any changes to data or transformation steps in your flow that cannot be pushed to the underlying data source are processed using the Data Engine process. For example, SQL Server does not natively support regular expressions. When connecting to SQL Server, Tableau Prep lets you write regular expression
calculations. Tableau Prep Conductor uses Data Engine to temporarily load the data and then perform the regular expression.

Performance and Scale Recommendations

- **Isolate flows to a separate node:** Running Tableau Prep Conductor on a separate node will isolate flow workflows from other Tableau workloads. This is highly recommended since Prep flows are CPU and RAM intensive.

- **Manage flow schedules:** You can control flow execution by creating flow schedules. These schedules let you determine when flows run, how frequently they run, the priority of that schedule, and whether to run items in that schedule serially or in parallel.

- **Add resources:** When scaling your Tableau Prep Conductor environment, we recommend scaling up to an 8 physical cores box per node running as many as 4 backgrounders on each. As you need more resources, we recommend adding more nodes to your server environment.

You can monitor user activity and performance of flows using Administrative views. For more information, see [Monitor Flow Health and Performance](#).

Topology and Configuration

By default, the installer enables Tableau Prep Conductor on all nodes that already have backgrounder enabled and where the following conditions are true:

- The Tableau Server has a valid Data Management product key.

- The node role is either set to allow all tasks or to allow only flows. For more information, on node roles, see the Node Roles in Server topic.

It is recommended that you enable Tableau Prep Conductor on a dedicated node to run flows. For more information, see the following topics:
For new Tableau Server installations, see Installing Tableau Server with Prep Conductor.

To enable Tableau Prep Conductor on an existing installation of Tableau Server, see Enable Tableau Prep Conductor to your Tableau Server Installation topic.

Enabling Tableau Prep Conductor on Tableau Server

To enable Tableau Prep Conductor, use the Tableau Data Management product key. Tableau Prep Conductor is now licensed and enabled at the server level.

Before you can start publishing flows to your Tableau Server, there are server-level and site-level settings you must configure or verify to prepare your Tableau Server to allow publishing, scheduling and monitoring flows.

Review the following topics to understand Tableau Prep Conductor licensing, and learn how to enable Tableau Prep Conductor:

- Enable and Configure Tableau Prep Conductor on Tableau Server: This topic provides step-by-step instructions on how to enable and configure Tableau Prep Conductor, and preparing your Tableau Server to allow publishing, scheduling, managing, and maintaining flows.

- License Overview: This topic describes licensing.

Enable and Configure Tableau Prep Conductor on Tableau Server

Tableau Prep Conductor is supported only on Tableau Server versions 2019.1 or later.

Tableau Prep Conductor is licensed through the Data Management Add-on, on a per Deployment basis, which is User-Based or Core-Based. A Deployment includes a licensed production Tableau Server installation and licensed non-production Tableau Server installations that support the production installation. For more information on Deployment, see the EULA Documentation.
For more information on how Tableau Prep Conductor licensing works, see Licensing Tableau Prep Conductor for Tableau Server.

Who can do this?

Server administrators can install Tableau Server and enable Tableau Prep Conductor.

Server-level settings can be configured by Tableau Server administrators, and site-level settings can be configured by Tableau Server and Site administrators.

Server Topology

When you install Tableau Server and enable Tableau Prep Conductor, using the Data Management product key, Tableau Prep Conductor is automatically enabled by default by the setup program.

For multi-node installations, by default, one instance of Tableau Prep Conductor is enabled on any node that has backgrounder installed. In the example below, Tableau Prep Conductor is enabled on node 2 and 3 where the backgrownders are also enabled, but not on node 1, 4, and 5.
Below is a visual representation of that work-flow:
Next step:

**New Tableau Server Installations: Step 1 (New Install): Install Tableau Server with Tableau Prep Conductor**

**Existing Tableau Server Installations: Step 1 (Existing Install): Enable Tableau Prep Conductor**

**Step 1 (New Install): Install Tableau Server with Tableau Prep Conductor**

This topic describes how to Tableau Prep conductor on a new installation of Tableau Server.

Tableau Prep Conductor is supported only on Tableau Server versions 2019.1 or later.

Tableau Prep Conductor is licensed through the Data Management Add-on, on a per Deployment basis. A Deployment includes a licensed production Tableau Server installation and...
licensed non-production Tableau Server installations that support the production installation. For more information on Deployment, see the EULA Documentation.

Who can do this?

Server administrators can install Tableau Server and enable Tableau Prep Conductor.

Before you install

The recommended topology for a production Tableau Server installation is a dedicated node for running flows. If you are currently planning to have a single node Tableau Server installation it is recommend that you add a second node and dedicate it to run flows.

- Review the hardware recommendations for Tableau Server and Tableau Prep conductor.
  - Minimum Hardware Requirements and Recommendations for Tableau Server installation on Windows.
  - Minimum Hardware Requirements and Recommendations for Tableau Server installation on Linux.

Install Tableau Server and enable Tableau Prep Conductor

Use the instructions provided in the following topics to install Tableau Server.

Windows: Install Tableau Server

Linux: Install Tableau Server topic

When you get to the Activate step, use the Tableau Server product keys to activate Tableau Server.

All product keys are available through the Customer Portal.

Configure public gateway settings

If your Tableau Server is set up with one of the following:
Load balancer to distribute requests across gateways.

Reverse proxy to authenticate external (internet) client requests and offloading SSL-based encryption.

You must configure the following public gateway settings:

```
tsm configuration set -k gateway.public.host -v <name>  # This should be the URL that your users are using to access Tableau Server
```

```
tsm configuration set -k gateway.public.port -v 443
```

For more information on configuring gateway settings, see Configuring Proxies for Tableau Server.

**Enable Tableau Prep Conductor**

*Use the following steps to add the Data Management product key to your Tableau Server:*

**Note:** This process requires a restart of the Tableau Server.

1. If the computer where you are running Tableau Server has been configured to connect to the internet through a forward proxy, follow the procedure in the topic, Configure Product Key Operations with Forward Proxy, before continuing.

2. Open TSM in a browser:

   `https://<tsm-computer-name>:8850`

3. Click Licensing on the Configuration tab and click Activate License:

4. Enter or paste your Data Management product key and click Activate:
5. On the Register page, enter your information into the fields and click **Register**.

6. Follow the prompts and restart Tableau Server after registration is complete.

**Verify Tableau Prep Conductor is enabled and running**

When you activate the Data Management product key, a single instance of Tableau Prep Conductor is automatically enabled on any node that has Backgrounder enabled.

**Use the following steps to verify that it is enabled and running:**

1. Open a browser and enter the Tableau Server URL, and append the dedicated TSM web UI port. Here are some examples of what the URL might look like:

   - https://localhost:8850/ (if you’re working directly on the server computer)
   - https://MarketingServer:8850/ (if you know the server’s name)
   - https://10.0.0.2:8850/ (if you know the server’s IP address)

   In the sign-in page that appears, enter your administrator user name and password.

   **Note:** Tableau Server creates and configures a self-signed certificate during the installation process. This certificate is used to encrypt traffic to the TSM Web UI. Because it’s a self-signed certificate, your browser will not trust it by default. Therefore, your browser will display a warning about the trustworthiness of the certificate before allowing you to connect.

2. In the Tableau Services Manager web interface, click the **Status** tab to see the status.

   - If Tableau Prep Conductor is enabled and running, you should see Tableau Prep Conductor in the list of processes as **Active** on at least one node. If Tableau Prep Conductor is not enabled, you will see Tableau Prep Conductor
Tableau Server on Linux Administrator Guide

in the list of processes, but with no status information for any of the nodes.

**Tableau Prep Conductor not enabled:**

```
<table>
<thead>
<tr>
<th>Process</th>
<th>node1</th>
<th>node2</th>
<th>node3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tableau</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Prep Conductor</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Node1</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Node2</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Node3</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```

**Tableau Prep Conductor enabled and running. In the image below**

**Tableau Prep Conductor is enabled on node1 and node3:**
Dedicate a node for Tableau Prep Conductor

On the node you are planning to dedicate to running flows, enable Backgrounder process if it is not already enabled. It is recommended that you do not run other processes like VizQL server on this node.

Because you are dedicating this node to running flows, you must configure Backgrounder to run only flow tasks. By default, the Backgrounder process runs tasks of all types, including flows, extract refreshes, and subscriptions. For more information, see Node Roles in Tableau Server.

Run the following tsm commands on that dedicated node to run only flow tasks:

1. Run the following command to allow Backgrounders on this node to run only flow tasks.

```
tsm topology set-node-role -n node1 -r flows
```
2. Set the node role on the initial node to no flows. The backgrounder on this node will run all jobs except flows:

```
  tsm topology set-node-role -n node1 -r no-flows
```

3. Apply the changes and restart Tableau Server:

```
  tsm pending-changes apply
```

**Multi-node installations**

If you have more than 2 nodes in your Tableau Server installation, you can choose to configure other nodes to run all tasks other than flows:

1. Restrict a node to not allow flows. This command removes Tableau Prep Conductor from this node and Backgrounders on this node will not run flow tasks.

```
  tsm topology set-node-role -n node1 -r no-flows
```

2. Apply the changes and restart Tableau Server:

```
  tsm pending-changes apply
```

Next step

**Step 2: Configure Flow Settings for your Tableau Server**

**Step 1 (Existing Install): Enable Tableau Prep Conductor**

This topic describes how to enable Tableau Prep conductor on your existing installation of Tableau Server.

Tableau Prep Conductor is supported only on Tableau Server versions 2019.1 or later. If you are using Tableau Server 2018.3 or earlier, you must first upgrade your Tableau Server to 2019.1 before enabling Tableau Prep Conductor on your Tableau Server installation.

Tableau Prep Conductor is licensed through the Data Management Add-on, on a per Deployment basis, which is User-Based or Core-Based. A Deployment includes a licensed
production Tableau Server installation and licensed non-production Tableau Server installations that support the production installation. For more information on Deployment, see the EULA Documentation.

This topic describes how to enable Tableau Prep conductor on your existing installation of Tableau Server.

Who can do this?

Tableau Server Administrators can install or upgrade Tableau Server, and enable Tableau Prep Conductor on Tableau Server.

Before you upgrade

Prepare for upgrade:

- Know before you upgrade
- Licensing Tableau Prep Conductor
- Tableau Server Hardware Requirements and Recommendations

Configure public gateway settings

If your Tableau Server is set up with one of the following:

- Load balancer to distribute requests across gateways.
- Reverse proxy to authenticate external (internet) client requests and offloading SSL-based encryption.

You must configure the following public gateway settings:

```
tsm configuration set -k gateway.public.host -v <name> (This should be the URL that your users are using to access Tableau Server)
```

```
tsm configuration set -k gateway.public.port -v 443
```
Tableau Server on Linux Administrator Guide

For more information on configuring gateway settings, see Configuring Proxies for Tableau Server.

Tableau Server Installations using User-Based licenses

The recommended topology for a production Tableau Server installation is a dedicated node for running flows. For more information, see Minimum Hardware Requirements and Recommendations for Tableau Server.

Tableau Server single-node installations

If you currently have a single node Tableau Server installation, it is recommended that you add a second node and dedicate it to running flows.

1. Run upgrade on your current Tableau Server installation using the information in the topics below:

   - Windows
   - Linux

   When you get to the Activate step, use the Tableau Server product keys to activate Tableau Server.

   All product keys are available through the Customer Portal.

2. After completing the installation, add the Data Management product key to enable Tableau Prep Conductor on your node. The Data Management product key, like your other server keys, are available through the Customer Portal.

   - In the Tableau Services Manager web interface, click Licensing on the Configuration tab and click Activate License.
   - Enter or paste your new product key and click Activate.
   - On the Register page, enter your information into the fields and click Register.

3. You will be prompted to restart the server. Restart the server and verify that Tableau
Prep Conductor is enabled and running.

- In the Tableau Services Manager web interface, click the **Status** tab to see the status. If Tableau Prep Conductor is enabled and running, you should see Tableau Prep Conductor in the list of processes as **Active**. If Tableau Prep Conductor is not enabled, you will see Tableau Prep Conductor in the list of processes, but with no status information.

### Tableau Prep Conductor not enabled:

![Tableau Prep Conductor not enabled](image1)

### Tableau Prep Conductor enabled and running:

![Tableau Prep Conductor enabled and running](image2)
Tableau Server on Linux Administrator Guide

4. Add a second node to your Tableau Server installation. The installer will enable certain required processes like the Cluster Controller. Enable Backgrounder process on it as it is required to run scheduled flow tasks. When you enable the Backgrounder process, the installer automatically enables a single instance of Data Engine and Tableau Prep Conductor on the node. Do not add any other processes on this node.

5. Run the following commands to dedicate this node to do only flow tasks. For more information on node roles, see Node Roles in Tableau Server.

   - Get the nodeID for your dedicated node to see the list of services on each node:
     
     tsm topology list-nodes -v.
     
   - Set the node role for the dedicated node using the nodeID that you got from running the command described above:
     
     tsm topology set-node-role -n <nodeID> -r flows.
     
   - Apply the changes, and restart the server:
     
     tsm pending-changes apply.
     
   - Review the status to ensure that all the processes are up and running and configured correctly:
     
     tsm status -v.

You have successfully added Tableau Prep Conductor to your Tableau Server installation.

Tableau Server multi-node installations

1. Run upgrade on your current Tableau Server Installation using the information in the topics below:

   - Windows

   - Linux
When you get to the **Activate** step, use the Tableau Server product keys to activate Tableau Server.

All product keys are available through the [Customer Portal](#).

2. After completing the installation, add the Data Management product key to enable Tableau Prep Conductor. Tableau Prep Conductor is automatically enabled on the nodes where you already have the Backgrounder process enabled. The Data Management product key, like your other server keys, are available through the [Customer Portal](#).

   - In the Tableau Services Manager web interface, click **Licensing** on the **Configuration** tab and click **Activate License**.

   - Enter or paste your new product key and click **Activate**.

   - On the **Register** page, enter your information into the fields and click **Register**.

3. You will be prompted to restart the server. Restart the server and verify that Tableau Prep Conductor is enabled and is running.

   - In the Tableau Services Manager web interface, click the **Status** tab to see the status of all the processes. If Tableau Prep Conductor is enabled and running, you should see Tableau Prep Conductor in the list of processes as **Active**. If Tableau Prep Conductor is not enabled, you will see Tableau Prep Conductor in the list of processes, but with no status information.

**Tableau Prep Conductor not enabled:**
Tableau Server on Linux Administrator Guide

### Tableau Prep Conductor enabled and running:

<table>
<thead>
<tr>
<th>Process</th>
<th>node1</th>
<th>node2</th>
<th>node3</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gateway</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Application Server</td>
<td>✓</td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Interactive Microservice</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>VoQD Server</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Cache Server</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cluster Controller</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Search &amp; Browse</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Exporter</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Non-Interactive Microservice</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Data Server</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Data Engine</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>File Store</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Repository</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td><strong>Tableau Prep Conductor</strong></td>
<td>●</td>
<td>●</td>
<td>●</td>
</tr>
<tr>
<td>Add Data</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Diagnostics Server</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>TSM Controller</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>License Server</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Status:**
- Active
- Backup
- degraded
- Error
- Stopped
4. Add a new node to your Tableau Server installation. The installer will enable certain required processes like the Cluster Controller. Enable Backgrounder process on it as it is required to run scheduled flow tasks. When you enable the Backgrounder process, the installer automatically enables a single instance of Data Engine and Tableau Prep Conductor on the node. Do not add any other processes on this node.

**Note:** The dedicated note counts towards to total count of the Coordination Service ensemble. You may need to deploy a Coordination Service on the new node depending on the total number of nodes you have in your cluster including the new dedicated node. For more information, see [Deploy a Coordination Service Ensemble](#).

5. Run the following command to dedicate this node to only doing flow related operations. For more information on node roles, see [Node Roles in Tableau Server](#).

   - Get the nodeID for your dedicated node to see the list of services on each node:
     - `tsm topology list-nodes -v`.
   - Set the node role for the dedicated node using the nodeID that you got from running the command described above:
     - `tsm topology set-node-role -n <nodeID> -r flows`.
   - Apply the changes and restart the server:
     - `tsm pending-changes apply`.
   - Review the status to ensure that all the processes are up and running and configured correctly:
     - `tsm status -v`.

6. At this stage, you may have Tableau Prep Conductor enabled on other nodes. By default, the Backgrounder process on a node performs all tasks of all types including flow tasks. To isolate Tableau Prep Conductor and flow tasks to only certain nodes, you can configure the Backgrounder to do one of the following:

   - **To run only flow tasks:** `tsm topology set-node-role -n <nodeID> -r flows`. 

---

Tableau Server on Linux Administrator Guide

1872 Version: 2019.4 Tableau Software
Tableau Server on Linux Administrator Guide

- To run all other tasks except flows: `tsm topology set-node-role -n <nodeID> -r no-flows`.

You have successfully added Tableau Prep Conductor to your Tableau Server installation.

Tableau Server Installations using Core-Based licenses

The recommended topology for a production Tableau Server installation is a dedicated node for running flows. For more information, see Minimum Hardware Requirements and Recommendations for Tableau Server.

The Data Management Add-on for Core-Based licenses includes product keys that enable Tableau Prep Conductor for your Tableau Server, and Tableau Prep Conductor cores that comes in units of four. The Tableau Prep Conductor cores should be applied to the node dedicated to running the flows. These product keys, like your other server keys, are available through the Customer Portal.

To learn more about Tableau Prep Conductor licensing, see Licensing Tableau Prep Conductor for Tableau Server.

Tableau Server single-node installations

If you currently have a single node Tableau Server installation, it is recommended that you add a second node and dedicate it to running flows.

1. Run upgrade on your current Tableau Server Installation using the information in the topics below:
   - Windows
   - Linux

2. Activate the product keys. This will enable Tableau Prep Conductor on the nodes where you already have the Backgrrounder process enabled.
3. You will be prompted to restart the server. Restart the server and verify that Tableau Prep Conductor is enabled and is running.

- In the Tableau Services Manager web interface, click the **Status** tab to see the status. If Tableau Prep Conductor is enabled and running, you should see Tableau Prep Conductor in the list of processes as **Active**. If Tableau Prep Conductor is not enabled, you will see Tableau Prep Conductor in the list of processes, but with no status information.

**Tableau Prep Conductor not enabled:**

![Tableau Services Manager screenshot](image)
Tableau Prep Conductor enabled and running:

4. Add a second node to your Tableau Server installation. The installer will enable certain required processes like the Cluster Controller. Enable Backgrounder process on it as it is required to run scheduled flow tasks. When you enable the Backgrounder process, the installer automatically enables a single instance of Data Engine and Tableau Prep Conductor on the node. Do not add any other processes on this node.

**Important:** The number of physical cores on this machine must be equal to, or less than the Tableau Prep Conductor cores you purchased. For example, if you purchased four Tableau Prep Conductor cores, your node can only have up to four physical cores. To understand about how Tableau Prep Conductor licensing works, see Licensing Tableau Prep Conductor for Tableau Server.

5. Run the following commands to dedicate this node to only doing flow tasks. For more information on node roles, see Node Roles in Tableau Server.

- Get the nodeID for your dedicated node to see the list of services on each node:
  
  ```
  tsm topology list-nodes -v.
  ```

- Set the node role for the dedicated node using the nodeID that you got from
running the command described above:

```bash
tsm topology set-node-role -n <nodeID> -r flows.
```

- **Apply the changes and restart the server:** `tsm pending-changes apply`.
- **Review the status to ensure that all the processes are up and running and configured correctly:**

```bash
tsm status -v.
```

You have successfully added Tableau Prep Conductor to your Tableau Server installation.

**Tableau Server multi-node installations**

1. Run upgrade on your current Tableau Server installation using the information in the topics below:
   - **Windows**
   - **Linux**

2. Activate the product keys. This will enable Tableau Prep Conductor on the nodes where you already have the Backgrounder process enabled.
   - In the Tableau Services Manager web interface, click **Licensing** on the **Configuration** tab and click **Activate License**.
   - Enter or paste your new product key and click **Activate**.
   - On the **Register** page, enter your information into the fields and click **Register**.

3. You will be prompted to restart the server. Restart the server and verify that Tableau Prep Conductor is enabled and is running.
In the Tableau Services Manager web interface, click the **Status** tab to see the status. If Tableau Prep Conductor is enabled and running, you should see Tableau Prep Conductor in the list of processes as **Active**. If Tableau Prep Conductor is not enabled, you will see Tableau Prep Conductor in the list of processes, but with no status information.

**Tableau Prep Conductor not enabled:**

![Status tab showing Tableau Prep Conductor not enabled](image)

**Tableau Prep Conductor enabled and running:**

![Status tab showing Tableau Prep Conductor enabled and running](image)
4. Add a new node to your Tableau Server installation. A dedicated node to run flow related operations is recommended for production Tableau Server installations. The installer will enable certain required processes like the Cluster Controller. Enable Backgrounder process on it as it is required to run scheduled flow tasks. When you enable the Backgrounder process, the installer automatically enables a single instance of Data Engine on the node. Do not add any other processes on this node.

**Note:** The dedicated note counts towards total count of the Coordination Service ensemble. You may need to deploy a Coordination Service on the new node depending on the total number of nodes you have in your cluster including the new dedicated node. For more information, see [Deploy a Coordination Service Ensemble](#).

**Important:**
The number of physical cores on this machine must be equal to, or less than the
Tableau Prep Conductor cores you purchased. For example, if you purchased four Tableau Prep Conductor cores, your node can only have up to four physical cores. To understand about how Tableau Prep Conductor licensing works, see Licensing Tableau Prep Conductor for Tableau Server.

5. Run the following commands to dedicate this node to only doing flow tasks. This will enable Tableau Prep Conductor on your new node. For more information, see Node Roles in Tableau Server.

   • Get the nodeID for your dedicated node to see the list of services on each node:
     
     tsm topology list-nodes -v.
   
   • Set the node role for the dedicated node using the nodeID that you got from running the command described above:
     
     tsm topology set-node-role -n nodeID -r flows.
   
   • Apply the changes and restart the server:
     
     tsm pending-changes apply.
   
   • Review the status to ensure that all the processes are up and running and configured correctly:
     
     tsm status -v.

6. At this stage, you may have Tableau Prep Conductor enabled on other nodes that have the Backgrounder process. By default, the Backgrounder process on a node performs all tasks of all types including flow tasks. To isolate Tableau Prep Conductor and flow operations to only certain nodes, you can configure the backgrounders to do one of the following:

   • To run only flow tasks:
     
     tsm topology set-node-role -n <nodeID> -r flows.
To run all other tasks except flows:

```bash
tsm topology set-node-role -n <nodeID> -r no-flows.
```

Next step

Step 2: Configure Flow Settings for your Tableau Server.

This topic describes the various flow settings that you can configure for your Tableau Server.

Who can do this?

Tableau Server Administrators can configure server and site level settings. Tableau Site Administrators can configure site level settings.

Publishing, Scheduling, and Credential Settings

When you activate Tableau Prep Conductor using the Data Management product key, Tableau Prep Conductor is enabled for the entire Tableau Server installation. You can further modify and customize the setting for sites.

Use the following instructions to configure settings related to flows for all your sites or for individual sites:

Use the following instructions to sign in to Tableau Server Admin pages:

- **Windows:** [Tableau Server Admin Pages](#).
- **Linux:** [Tableau Server Admin Pages](#).

**Configure whether publishing and scheduling flow should be allowed for a site:**

1. **Enable users to publish and schedule flows:** This setting is enabled by default when you enable Tableau Prep Conductor. If you have multiple sites, you can
selectively turn off Tableau Prep for Server for individual sites. If you disable this setting for a site that once allowed flows, see Implication of disabling Tableau Prep Conductor to learn more.

On the **General** page under **Settings**, scroll to the **Flows** setting and clear the **Allow user to publish and schedule flows** check box to turn off Tableau Server for the site.

![General Settings](image)

2. **Embed Credentials**

- **Allow publishers to embed credentials in a data source, flow or workbook**: This setting allows publishers to attach passwords to published flows that will automatically authenticate web users.

- **Allow publishers to schedule flow runs and data extract refreshes**: This option is only available if setting above is enabled. When this setting is enabled, publishers will see scheduling options in the Publish dialog box.
Implication of disabling Tableau Prep Conductor

If you disable Tableau Prep Conductor after using it for a while, you will not be able to see the flows, schedules, tasks, and other things related to flows. The following table gives you more information on what you can and cannot see when you disable Tableau Prep for Server completely or only for specific sites:

<table>
<thead>
<tr>
<th></th>
<th>Prep not enabled at Server level</th>
<th>Prep enabled at Server level, but disabled for a site</th>
<th>Prep enabled for both Server and site</th>
</tr>
</thead>
<tbody>
<tr>
<td>Show flows</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Show tasks/schedules in Server view</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Show tasks/schedules in Site view</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
</tr>
<tr>
<td>Show Site setting (only for Server Admins)</td>
<td>Yes (disabled)</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Show TSM status</td>
<td>Yes (Tableau Prep Conductor is not shown)</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Show TSM settings</td>
<td>Yes (disabled)</td>
<td>Yes</td>
<td>Yes</td>
</tr>
</tbody>
</table>

**Important:** Scheduled tasks will continue to run even when Tableau Prep Conductor is disabled for that site, but will fail.

Configure notifications for flow failures

You can configure Tableau Server to send email notifications for flow run failures. The notifications are sent for failures that occur when running the flows through either a scheduled
task or a manual run using the **Run now** menu option. You must first enable the server-wide setting, and then configure at the site level.

**To enable the server-wide email notification**

You can either use the Tableau Services Manager (TSM) web interface or TSM CLI as described below:

**Use the TSM web interface**

1. Open TSM in a browser:
   
   https://<tsm-computer-name>:8850.

2. Click **Notifications** on the **Configuration** tab and click **Email Server**.

3. Enter the email server information.

4. Click the **Events** tab.

5. Under **Content Updates**, select **Send emails for extract refresh failures and flow run failures** if not already turned on by default.

6. Click **Save Pending Changes** after you've entered your configuration information.

7. Click **Apply Changes and Restart**.

**Use the TSM CLI**

The notification values can be set individually with the tsm configuration set command:

Windows: `tsm configuration`.

---

Tableau Software  
Version: 2019.4  
1883
Linux: tsm configuration.

Set notification values

Use the tsm configuration set command with the following syntax to enable flow failure notifications, run the following command:

```bash
tsm configuration set -k backgrounder.notifications_enabled -v true
```

**Note:** This will enable email notification for both extract refresh failures and flow failures.

After you are done setting values, you must run the following command:

```bash
tsm pending-changes apply
```

The pending-changes apply command displays a prompt to let you know this will restart Tableau Server if the server is running. The prompt displays even if the server is stopped, but in the case there is no restart. You can suppress the prompt using the --ignore-prompt option, but this does not change the restart behavior.

To configure email notification for a site:

On the General page under Settings, scroll to the Flow Run and Refresh Failure Notifications setting and clear the Send email to data source, flow, and workbook, when flow runs and scheduled refreshes fail check box to turn off refresh failure and flow run notifications.
Next step

Step 3: Create Schedules for Flow Tasks

Who can do this?

Tableau Server Administrators can create and modify schedules. Schedules are created at the server level and apply across all the sites on a server.

Create a new schedule:

1. On the Schedules tab, click New Schedule.

2. Enter the following information in the New Schedule dialog box and click Create.
   
   - **Name**: Enter a descriptive name for the schedule. Typically, this includes the description of the schedule frequency.
   
   - **Task Type**: Select Flow as the task type.
• **Default Priority**: You can define a default priority from 1 to 100, where 1 is the highest priority. This value will be assigned to the tasks by default. If two tasks are pending in the queue, the one with the higher priority runs first.

• **Execution**: Choose whether a schedule will run in parallel or serially. Schedules that run in parallel run on all available backgrounder processes so that they can complete faster.

• **Frequency**: You can define an hourly, daily, weekly, or monthly schedule.

Next step

Step 4: Safe list Input and Output locations

Step 4: Safe list Input and Output locations

This topic describes the rules that apply to this feature and how to safe list the directories on your network.
Flow input and output connections may need to connect to databases or files in the directories on your network. You must safe list the directories you want to allow access to. Input and Output connections will only be allowed to connect to data in the safe listed locations. By default, no connections are allowed. You can still publish the flows and any data that is embedded in the flow file (tflx) to Tableau Server.

Who can do this?

Tableau Server Administrators who also are a member of the tsmadmin group can configure settings using tsm commands.

How to safe list input and output locations

The following rules apply and must be considered when configuring this setting:

- The directory paths should be accessible by Tableau Server. These paths are verified during server startup and at flow run time and are not verified at the time of publishing the flow to Tableau Server.

- Network directory paths have to be absolute and cannot contain wildcards or other path traversing symbols. For example, `\\myhost\myShare\*` or `\\myhost\myShare*` are invalid paths and would result in all the paths as disallowed. The correct way to safelist any folder under myShare would be `\\myhost\myShare` or `\\myhost\myShare\`. 

  **Note:** The `\\myhost\myShare` configuration will not allow `\\myhost\myShare1`. In order to safe list both of these folders safe list them as `\\myhost\myShare;\\myhost\myShare1`.

- **Windows:**

  - The value can be either `*`, (for example, `tsm configuration set -k maestro.input.allowed_paths -v "*"`) to allow any network
directory, or a specified list of network directory paths, delimited by a semicolon (;).

- If the path contains spaces or special characters you will have to either use single or double quotes. Whether you use single or double quotes depends on the shell that you are using.

- No local directory paths are allowed even when the value is set to *.

**Linux:**

- The value can be either *, (for example, `tsm configuration set -k maestro.input.allowed_paths -v "*"`) meaning that any path, including local (with the exception of some system paths configured using "native_api.internal_disallowed_paths"), or a list of paths, delimited by a semicolon (;).
- You must be using a kernel version of equal to or later than 4.7. Safe listing is not supported on kernel version earlier than 4.7. To check the kernel version, in the Linux terminal, type the command `uname -r`. This will display the full version of the kernel you are running on the Linux machine.

**Note:** If a path is both on the flows allowed list and internal_disallowed list, internal_disallowed takes precedence.

Use the following commands to create a list of allowed network directory paths:

**For input connections:**

```
  tsm configuration set -k maestro.input.allowed_paths -v your_networkdirectory_path_1;your_networkdirectory_path_2
  tsm pending-changes apply
```

**For output connections:**

```
  tsm configuration set -k maestro.output.allowed_paths -v your_networkdirectory_path_1;your_networkdirectory_path_2
```
Important:
These commands overwrite existing information and replace it with the new information you provided. If you want to add a new location to an existing list, you must provide a list of all the locations, existing, and the new one you want to add. Use the following commands to see the current list of input and output locations:

```
tsm configuration get -k maestro.input.allowed_paths
tsm configuration get -k maestro.output.allowed_paths
```

Next step

**Step 5: Optional Server Configurations**

The options described in this topic are not required to enable flow publishing and scheduling flows on Tableau Server. They can be used to customizing your environment according to your requirements.

Who can do this?

Tableau Server administrators can make changes to server configurations.

Set the timeout period for flows

You can set time limits for how long a flow can run to make sure that subsequent tasks are not held up due to stalled tasks. The following two `tsm` command options determine how long a flow task can run before the flow background task is canceled. These two commands together determine the total timeout value for flow tasks.

Use the following `tsm` commands to change the default timeout period of 4 and half hours:

```
backgrounder.extra_timeout_in_seconds
```
(Default value: 14400 seconds or 4 hours)

```
backgrounder.default_timeout.run_flow
```

(Default value: 1800 seconds or 30 minutes)

Set the threshold for suspended flow tasks

By default, a flow task is suspended after 5 consecutive flow task failures. To change the threshold number of flow task failures that can occur before they are suspended, use the following tsm configuration set command:

```
tsm configuration set -k backgrounder.flow_failure_threshold_for_run_prevention -v <number>
```

This sets the threshold for the number of consecutive failed flow tasks necessary before suspending the tasks. This is a server-wide setting.

**About Tableau Catalog**

Data is increasing in volume, formats, and importance leading to more complex environments. With the rapid pace that data changes, it can be hard to keep track of that data and how it’s being used in such complex environments. At the same time, more users need to access more of that data in more places, and it’s difficult for users to find the right data. Ultimately, this causes a lack of trust in the data because people question whether they’re using the right source or if it’s up to date.

Tableau Catalog integrates features like lineage, impact analysis, data dictionary, data quality warnings, and search into your Tableau applications, helping solve these problems differently from a stand-alone catalog. It focuses on both IT and the end user so that everyone using Tableau Server or Tableau Online has more visibility and trust into the data, while also enabling more discoverability. Tableau Catalog builds a catalog out of the Tableau content being used by your organization, enabling comprehensive functionality like the following:
Impact analysis and lineage.
- You can see the workbooks and other Tableau content that depend on particular columns or fields from tables or data sources you manage. When you need to make changes to your data, you can notify the impacted Tableau authors using email.
- As a workbook author, you can use lineage to trace the fields that your workbook depends on.
- As a user, when you use a Tableau visualization, you can see where the data came from that was used to create the view.

Curation and trust. As a data steward, you can add helpful metadata, like descriptions and certification, so that users find the right data. You can set data quality warnings, view data details on the Data Details pane, certify assets, and remove assets from the catalog.

Data discovery. You can use Tableau Catalog to search for databases, tables, and data sources to analyze in Tableau and connect to them from the search results.

Starting in 2019.3, Tableau Catalog is available in the Data Management Add-on to Tableau Server and Tableau Online. When the product key is active and enabled, the catalog features described above are integrated into the product you’re using, so you can work with the data where you find it.

How Tableau Catalog works

Tableau Catalog discovers and indexes all the content on your site—workbooks, data sources, sheets, and flows—to gather metadata about the content. From the metadata, databases and tables (also referred to as external assets) are identified. Knowing the relationships between the content and the external assets enables Tableau to display the lineage of the content and assets. Tableau Catalog also enables users to connect to external assets using Tableau Server or Tableau Online.

For information about how you can use Tableau Catalog to support data governance in your organization, see Governance in Tableau in the Tableau Blueprint Help.
Key Tableau Catalog terms

- Metadata. Information about the data.
- Tableau content. Content created in Tableau such as workbooks, data sources, and flows.
- External assets. The metadata about the databases and tables used by the Tableau content published to Tableau Server or Tableau Online.

License Tableau Catalog

Tableau Catalog is licensed through the Data Management Add-on. For information about how Data Management Add-on licensing works, see License the Data Management Add-on.

Enable Tableau Catalog

After Tableau Server or Tableau Online is licensed with the Data Management Add-on, you can enable Tableau Catalog by doing one of the following tasks:

- **For Tableau Online**, no action necessary. Tableau Catalog is on by default, configured to use derived permissions, and ready to use. For more information about derived permissions, see the Permissions on assets and their metadata topic.

- **For Tableau Server**, the Server admin must first enable the Tableau Metadata API using the `tsm maintenance metadata-services command`. For more information, see Enable Tableau Catalog.

  After the Metadata API is enabled, Tableau Catalog is on by default, configured to use derived permissions, and ready to use. For more information about derived permissions, see the Permissions on assets and their metadata topic.

Features and functionality

To learn more about the features you can use with Tableau Catalog, see the following Help articles:
Data discovery

- In the **Connect** pane on Tableau Desktop, under **Search for Data** select **Tableau Server** to connect to data using Tableau Server or Tableau Online. When Tableau Catalog is enabled, in addition to searching for published data sources to connect to, you can now search for and connect to the specific databases and tables used by published data sources and workbooks on your Tableau Server or your Tableau Online site.
- **Search** is expanded to include results based on fields, columns, databases, and tables when Tableau Catalog is enabled.
- If you author in the web, you can also connect to databases and tables, as well as published data sources, from **Tableau Server** and **Tableau Online**.

Curation and trust

- **Certify your data assets** to help users find trusted and recommended data.
- Set **data quality warnings** to alert users to data quality issues, such as stale or deprecated data.
- Better understand published visualizations by using the **Data Details** tab to see information about the data used.
- **Add descriptions** to databases, tables, and columns to help users find the data they're looking for.

Lineage and impact analysis

- Use **lineage** to trace the source of your data and to analyze the impact of changes to your data and identify which users might be impacted.
- **Email owners** of a workbook, data source, or flow or contacts for a database or table about data-related updates.

Developer resources

You can use the **Tableau Metadata API** to programatically query metadata from the content published to Tableau Server or Tableau Online. The Metadata API is fast and flexible and is best when you are looking to find out specific information about the relationship between metadata and its structures.
In addition to the Metadata API, you can use metadata methods in the Tableau Server REST API to programmatically update certain metadata. For more information about metadata methods, see Metadata Methods in the Tableau Server REST API.