Tableau Server on Windows Administrator Guide
Plan Your Deployment

Tableau Server on Windows: Start Here ................................................................. 2

Server Administrator Overview ............................................................................. 3

Tableau Server on Windows: Start Here

Tableau Server® on Windows by Tableau Software lets Tableau users share workbooks and extracts with other users in your organization.

Use the links below to get started using Tableau Server on Windows.

Plan

<table>
<thead>
<tr>
<th>Server Administrator Overview</th>
<th>What's New in Tableau Server</th>
</tr>
</thead>
<tbody>
<tr>
<td>Infrastructure Planning</td>
<td>Security</td>
</tr>
<tr>
<td>Planning a Site</td>
<td></td>
</tr>
</tbody>
</table>

Deploy

<table>
<thead>
<tr>
<th>Configure Tableau Server</th>
<th>Install Tableau Server in the AWS Cloud</th>
</tr>
</thead>
<tbody>
<tr>
<td>Distributed Environments</td>
<td></td>
</tr>
</tbody>
</table>

Manage Server

<table>
<thead>
<tr>
<th>tabadmin configuration options</th>
<th>Extract Refresh Schedules</th>
</tr>
</thead>
<tbody>
<tr>
<td>Add or Update Sites</td>
<td>Add Users to the Server</td>
</tr>
</tbody>
</table>

Manage Sites
Tableau Server on Windows 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.

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

Looking for Tableau Server on Linux? See Server Administrator Overview.
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. Extracts can be resource intensive. If your organization plans to make heavy use of extracts, review the topic, Optimize for Extracts.
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. You can optimize performance for live data connections with caching configurations and specifying initial SQL commands.

**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. In 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

- 8 -
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.

  Permissions required for the Tableau Server administrator page are role based. The roles are generated and managed by Tableau Server.

- **Tableau Server Configuration Utility**: This is the primary utility for server-wide configurations. The configurations made with this utility 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 the utility. You must use an account with local server administrator permissions to run the Tableau Server Configuration Utility.

  **tabadmin**: This is a command line tool that allows you to configure multiple components of Tableau Server. tabadmin installs with Tableau Server by default.

  **tabcmd**: You can use the tabcmd command-line utility on a Windows computer to create scripts to automate administrative tasks on your Tableau Server site. 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.

The New Data Engine in Tableau Server

Hyper is Tableau’s new in-memory Data Engine technology optimized for fast data ingests and analytical query processing on large or complex data sets. Hyper powers the new Data Engine in Tableau Server 10.5, Tableau Desktop 10.5, Tableau Online, and Tableau Public. The Data Engine in 10.5 is used when creating, refreshing or querying extracts. It is also used for cross-database joins to support federated data sources with multiple connections.

In this article

- Performance benefits
- Memory and CPU usage
- Changes to Server configuration, topology, and extracts

Performance benefits

Starting in 10.5, Hyper technology is 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 the new 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.

Here are some reasons why the new Data Engine 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 new 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.
Memory and CPU usage

The new 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 new Data Engine monitors itself to stay within the limits set in the 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 new 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 (ex: 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 10.5.
Memory usage of the new Data Engine depends on the 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, the Data Engine shifts to spooling by writing to disk. Spooling, therefore 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).

Changes to Server configuration, topology, and extracts

This section describes changes introduced in Tableau Server 10.5 with the new Data Engine.

Hardware requirements and recommendations

There are no changes to hardware requirements and recommendations. For the latest hardware requirements and recommendations, see Minimum Hardware Requirements and Recommendations for Tableau Server.

Licensing

There are no changes to licensing requirements or trial versions of Tableau Server. For the latest information on licensing, see Licensing Overview.

Server configuration and topology

- In versions 10.4 and earlier, it was possible to configure more than one instance of Data Engine on a node. Beginning in Tableau Server 10.5, 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. You can no longer configure multiple instances of Data Engine on a node. The new 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 versions 10.4 and earlier, for distributed installations, the backgrounder process launched temporary instances of Data Engine (tdeserver64.exe) process during extract creation and refreshes. 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. For more information on configuration recommendations, see Distributed Installation Recommendations.

**Scalability with the new Data Engine:**

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.
Extracts

Beginning with version 10.5, new extracts use the new .hyper format. However your existing extracts will not automatically be upgraded to the new .hyper format when you upgrade your Tableau Desktop or Tableau Server. The new Data Engine can read both the .tde and .hyper file formats, so you can continue to view workbooks that have .tde extracts. However, when certain extract tasks are performed on the .tde extract, the extract is then upgraded to a .hyper format. An automatic refresh will also upgrade the .tde extract to a .hyper format.

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 Pre-Upgrade Tasks.

Use the Extract API 2.0 to create (.hyper) extracts. Extract API 2.0 requires Tableau 10.5 and later. If you are using Tableau 10.4 and earlier, use the Tableau SDK to create .tde extracts. For more information, see Extract API 2.0.

Note: Tableau SDK is no longer being maintained.

Extract File Size

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.

Version compatibility between Tableau Desktop and Tableau Server

Upgraded extracts are not backward compatible: Once the extracts are upgraded from .tde to .hyper, they cannot be opened or refreshed in Tableau Desktop versions earlier than 10.5. We recommend that Tableau Server and all the Tableau Desktops in your organization be on the same version and be upgraded to 10.5 on the same schedule.
You may not be able to do some tasks when the Tableau Server and Tableau Desktop are on different versions. For example:

- You cannot publish extracted data sources (.hyper) from Tableau Desktop 10.5 to a Tableau Server 10.4 or earlier.
- You cannot download and open an upgraded .hyper extract on a 10.4 Tableau Desktop.
- You cannot use the downgrade and publish feature with a Hyper extract.

For more information and details, see Impact of extract upgrade.

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.

Installing Tableau Server

If you are installing a new instance of Tableau Server (not performing an upgrade), you do not need to make any special considerations for the new Data Engine. See Run Server Setup.

Upgrading to Tableau Server 10.5

You can upgrade previous versions of Tableau Server to Tableau Server 10.5 on the Windows platform. Review the upgrade information below.

- You will also need a new version of tabcmd to publish the upgraded .hyper extracts. For more information, see tabcmd.
- Generally there are no changes to the server upgrade experience for the new Data
Engine. If you are upgrading a multi-node server, you will be prompted to manually upgrade the additional nodes. If you are running a previous Windows version of Tableau Server, the upgrade may affect your current configuration for a multi-node cluster, especially if you have certain processes isolated on a separate node. For example, if you upgrade a node that has just the Backgrounder installed, after upgrade you will have one instance of Data Engine installed on that node as well.

- For more information on how to perform the upgrade on a single-node server, see Upgrade a Single-Node Tableau Server.

- For more information on how to perform the upgrade on a multi-node server, see Upgrade a Multi-Node (Distributed) Tableau Server.
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.

Requirements

In addition to the following requirements, you can find technical specifications for Tableau Server on the Tableau web site.


- **Supported browsers**—Tableau Server 10 supports Internet Explorer 11 in native mode, and the latest versions of Chrome, Firefox, and Safari.

This has potential to impact:

- Customers installing Tableau Server for the first time on Windows 8 or Windows Server 2012 (non-R2). For more information, see Internet Explorer
Support.

- Customers accessing embedded Tableau views in web pages that force Internet Explorer into compatibility mode. For more information, see Internet Explorer Compatibility Mode.

- Minimum requirements—The computer you install Tableau Server on must meet or exceed the minimum hardware requirements. Tableau Server will not install if your computer does not meet the minimum requirements.
  - Minimum requirements are appropriate for testing and prototyping.
  - For production environments your computers should meet or exceed the minimum recommendations.

For more information, see Minimum Hardware Requirements and Recommendations for Tableau Server.

- Additional disk space requirements — 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.

- Administrative account—The account under which you install Tableau Server must have permission to install software and services.

- 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. For more information, see Update Configuration for a New IP Address.

Optional and recommended considerations

The following technical specifications may not apply to all scenarios.

- Optional: Run As User account—A Run As User account for the Tableau Server service to run under is useful if you’re using NT Authentication with data sources or if you’re planning on doing SQL Server impersonation. For more information, see Run
As User and SQL Server Impersonation.

In some organizations, Group Policy or other system management solutions are used to standardize permissions and accounts on application servers. If your organization runs a such a solution, be sure to configure the system to accommodate the folder permissions required by the Run As User account. See Verify Folder Permissions.

- **IIS and port 80**—Our recommendation is to run Tableau Server on a dedicated computer. Running Tableau Server on a computer with other applications may degrade performance. In addition, running Tableau Server with other applications may interfere with ports required by Tableau Server. Tableau Server's gateway listens on port 80, which is also used by Internet Information Services (IIS) by default. If are installing Tableau Server on a computer that's also running IIS, you should modify the Tableau's gateway port number to avoid conflict with IIS. See Tableau Server Ports and Edit the Default Ports for details.

- **Antivirus software**—Antivirus software that performs full drive scans can interfere with Tableau Server. In some cases, interference from antivirus software will prohibit Tableau Server from starting up. In addition, full scans of active drives where Tableau Server is installed may impact performance. If you plan to run antivirus software on the computer running Tableau Server, follow the recommendations in the Knowledge Base.

**Ports and networking**

By default Tableau Server requires several TCP/IP ports to be available to the server. See the topic Tableau Server Ports for the full list, including which ports must be available for all installations vs. distributed installations or failover-ready installations. The default ports can be changed if there is a conflict. See Edit the Default Ports to learn how.

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. To provide internet access to Tableau Server we recommend configuring a proxy server. See Configuring Proxies for Tableau Server.
Installation directory

By default, Tableau Server will install on the system drive. The drive where Windows is installed is the system drive. In most cases, the system drive is the C:\ drive. In this default case, Tableau Server will install into the following directories:

- C:\Program Files\Tableau\Tableau Server
- C:\ProgramData\Tableau\Tableau Server

Non-default installation locations

Rather than install onto the system drive, some organization install applications on a separate drive. During setup (or with automated installation), you can specify a different installation location. If you select a different installation drive or folder location during setup, then the data directory for Tableau Server will install into the same path. This means that if you install to a non-default location, then the \ProgramData\Tableau\Tableau Server path will not be created.

For example, installing Tableau Server to the root of a non-system drive (e.g., D:\), results in a single install path: D:\Tableau\Tableau Server.

When you install to a non-default location, you must take the following into consideration:

- You may need to configure Run As User permissions manually. If you do not use the predefined local account, NetworkServices, as the Run As User account, then you will need to set permissions for the account that you will be using. You must set these permissions after you install Tableau Server. See Verify Folder Permissions.

- Log files are stored at <installation path or drive>\Tableau\Tableau Server\logs.

- C:\ProgramData\Tableau\Tableau Server\data path in default installation is converted to <installation path or drive>\Tableau\Tableau Server\data.
• You will need to specify the same location when you upgrade to newer versions of Tableau Server. See Upgrade Tableau Server to a Non-Default Location.

Drivers

You may need to install additional database drivers. Download drivers from www.tableau.com/support/drivers.

Configuration Information

When you install and configure Tableau Server you may be asked for the following information:

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
<th>Your Information</th>
</tr>
</thead>
<tbody>
<tr>
<td>Run As User (service account)</td>
<td>The server must have a user account that the service can use. The default is the built-in Windows Network Service account. If you use a specific user account you'll need the domain name, user name, and password.</td>
<td>Username:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Password:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Domain:</td>
</tr>
<tr>
<td>Active Directory</td>
<td>Instead of using Tableau’s built-in user management system, you can authenticate through Active Directory. In this scenario, Tableau Server must be installed in a domain in Active Directory. You’ll need the fully-qualified domain name for the domain where user accounts are managed.</td>
<td>Active Directory</td>
</tr>
<tr>
<td>Open port in Windows firewall</td>
<td>When selected Tableau Server will open the port used for http requests in the Windows Firewall software to allow other machines on your network to access the server.</td>
<td>__ - Yes</td>
</tr>
<tr>
<td></td>
<td></td>
<td>__ - No</td>
</tr>
</tbody>
</table>
What's New and What's Changed

Find out about the new and changed features in Tableau Server:

- See the What's New in Tableau Server topic in the Tableau Server online help for information about key new features.
- See What's Changed - Things to Know Before You Upgrade for information about changes that may impact your users.

Licensing Overview

Tableau Server licenses have two aspects: the license model (term or perpetual) and the license metric (user-based or core-based). Term licenses are user-based, and perpetual licenses usually core-based.

License model: term or perpetual

Tableau Server can be licensed under two models: term licensing and a perpetual licensing. Term licenses, also known as subscription licenses, allow you to use and update Tableau Server for a specified period of time.

Perpetual licenses do not expire, so you can continue to use Tableau Server as long as you want. However, to get access to product updates and technical support you must purchase Support and Maintenance services.

License metric: user-based or core-based

In addition to the license model, your license is also 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.

## Tableau Server Processes

Tableau Server installs a number of processes that work together to deliver the features that make up Tableau Server.

In this article

- Configuring processes
- Licensed processes
- List of processes

Configuring processes

Certain processes listed below cannot be configured: cluster controller and coordination service are installed on every node as part of the base install. They are required on every server node and do not count against a core-based license. Data engine is installed when you install file store, Vizportal, VizQL Server, data server, or backgrounder, and cannot be installed separately. Every node that has an instance of one of those processes will also have a single instance of the data engine process.

The topics Performance Tuning Examples and High Availability describe some of the approaches you can take when configuring processes. High-level status for each process is displayed on the server's Status page and more detailed information related to some of the processes—such as the background process—is in the Administrative Views topic.

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 primary Tableau Server computer to verify they are licensed. If they cannot confirm there is a valid license, for example, if the primary 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.
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.

List of processes

For information on log files generated by these processes, see Server Log File Locations.

<table>
<thead>
<tr>
<th>Process</th>
<th>Purpose</th>
<th>Multi-Threa-</th>
<th>Performance Characteristics</th>
<th>Licensed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Application Server</td>
<td>Handles the web application, REST API - calls, supports browsing and searching</td>
<td>Yes</td>
<td>Only consumes noticeable resources during infrequent operations, like publishing a workbook with an extract, or generating a static image for a view. Its load can be created by browser-based interaction and by tabcmd.</td>
<td>Yes</td>
</tr>
<tr>
<td>vizportal.exe</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Backgrounder</td>
<td>Executes server tasks, including extract refreshes, subscriptions, 'Run Now' tasks, and tasks initiated from tabcmd</td>
<td>No</td>
<td>A single-threaded process where multiple processes can be run on any or all machines in the cluster to expand capacity. The backgrounder normally doesn't consume much process memory, but it can consume CPU, I/O, or network resources based on the nature of the workload presented to it. For example, performing large extract refreshes can use network bandwidth to retrieve data. CPU resources can be consumed by data retrieval or complex tabcmd tasks.</td>
<td>Yes</td>
</tr>
<tr>
<td>backgrounder.exe</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Process</td>
<td>Purpose</td>
<td>Multi-Threaded</td>
<td>Performance Characteristics</td>
<td>Licensed</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>--------------------------------</td>
<td>----------------</td>
<td>---------------------------------------------------------------------------------------------</td>
<td>----------</td>
</tr>
<tr>
<td>Cache Server redis-server.exe</td>
<td>Query cache</td>
<td>No</td>
<td>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 API server and application server to a lesser extent) make cache requests to 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>No</td>
</tr>
<tr>
<td>Cluster Controller cluster-controller.exe</td>
<td>Responsible for monitoring various components, detecting failures, and executing failover when needed</td>
<td>n/a</td>
<td>Included in the base install on every node.</td>
<td>No</td>
</tr>
<tr>
<td>Coordination Service zookeeper.exe</td>
<td>In distributed install-</td>
<td>n/a</td>
<td>Always installed on the primary node. For server installations with three to five nodes, also installed on the first two worker nodes. For server install-</td>
<td>No</td>
</tr>
<tr>
<td>Process</td>
<td>Purpose</td>
<td>Multi-Threaded</td>
<td>Performance Characteristics</td>
<td>Licensed</td>
</tr>
<tr>
<td>------------------</td>
<td>-------------------------------------------------------------------------</td>
<td>----------------</td>
<td>----------------------------------------------------------------------------------------------</td>
<td>----------</td>
</tr>
<tr>
<td>Data Engine</td>
<td>Creates extracts and processes queries.</td>
<td>Yes</td>
<td>The data engine’s workload is generated by requests from the VizQL server, application server, API server, data server, and backgrounder server processes. The data engine services requests from most of the other server processes as well. It is the component that loads extracts into memory and performs queries against them, and writes the extracts to the database. The data engine is multi-threaded to handle multiple requests at a time. For more information about CPU and memory usage, see The New Data Engine in Tableau Server. The data engine is automatically installed when you install file store,</td>
<td>Yes</td>
</tr>
<tr>
<td>Process</td>
<td>Purpose</td>
<td>Multi-Threa-</td>
<td>Performance Characteristics</td>
<td>Licensed</td>
</tr>
<tr>
<td>--------------------</td>
<td>-------------------------------------------------</td>
<td>------------</td>
<td>---------------------------------------------------------------------------------------------</td>
<td>----------</td>
</tr>
<tr>
<td>Data Server</td>
<td>Manages connections to Tableau Server data sources</td>
<td>Yes</td>
<td>VizQL Server, Application Server (VizPortal), Data Server, or Back-grounder. You cannot install Data Engine separately.</td>
<td>Yes</td>
</tr>
<tr>
<td>dataserver.exe</td>
<td></td>
<td></td>
<td>Because it’s a proxy, it’s normally only bound by network, but it can be bound by CPU with enough simultaneous user sessions. Its load is generated by browser- and Tableau Desktop-based interaction and extract refresh jobs for Tableau Server data sources.</td>
<td></td>
</tr>
<tr>
<td>File Store</td>
<td>Automatically replicates extracts across data engine nodes</td>
<td>n/a</td>
<td>A data engine is automatically installed when file store is installed, unless the node already has an instance of data engine.</td>
<td>No</td>
</tr>
<tr>
<td>filestore.exe</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Repository</td>
<td>Tableau Server database, stores work-book and user metadata</td>
<td>n/a</td>
<td>Normally consumes few resources. It can become a bottleneck in rare cases for very large deployments (thousands of users) while performing operations such as viewing all workbooks by user or changing permissions. For more information, see Tableau Server Repository.</td>
<td>No</td>
</tr>
<tr>
<td>postgres.exe</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Search &amp; Browse</td>
<td>Handles</td>
<td>Yes</td>
<td>The process is memory bound first,</td>
<td>No</td>
</tr>
<tr>
<td>Process</td>
<td>Purpose</td>
<td>Multi-Threaded</td>
<td>Performance Characteristics</td>
<td>Licensed</td>
</tr>
<tr>
<td>------------------------</td>
<td>-------------------------------------------------------------------------</td>
<td>----------------</td>
<td>---------------------------------------------------------------------------------------------</td>
<td>----------</td>
</tr>
<tr>
<td>search-server.exe</td>
<td>fast search, filter, retrieval, and display of content metadata on the server</td>
<td></td>
<td>and I/O bound second. The amount of memory used scales with the amount of content (number of sites/-projects/workbooks/datasources/views/users) on the server.</td>
<td></td>
</tr>
<tr>
<td>VizQL Server vizqlserver.exe</td>
<td>Loads and renders views, computes and executes queries</td>
<td>Yes</td>
<td>Consumes noticeable resources during view loading and interactive use from a web browser. Can be CPU bound, I/O bound, or network bound. Process load can only be created by browser-based interaction. Can run out of process memory.</td>
<td>Yes</td>
</tr>
</tbody>
</table>

Infrastructure Planning

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 identity store. 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 Active Directory, SAML, or OpenID. In all cases, whether authentication takes place locally or is external, each user identity must be represented in the Tableau Server identity store, which is managed by the repository.

Looking for Tableau Server on Linux? See Authentication.

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.

User identity in Tableau Server

When you install Tableau Server, you must select the process that the server will use to manage user authentication: local authentication or Active Directory. Before you install Tableau Server, you should understand how these two options impact your overall authentication strategy. After you select and set the authentication process, Tableau Server will configure the various components for the authentication method that you have selected. After this configuration is complete, you cannot change the authentication method. In fact, to change this configuration, you must uninstall the server, delete the configuration on the computer, and then reinstall the server.
Local authentication

If the server is configured to use local authentication, then the Tableau Server identity store is used exclusively to authenticate 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”).

Use local authentication if any of the following are true:

- Your organization does not manage users with Active Directory
- You do not want to use Active Directory
- You want to use OpenID for authentication and single sign-on

When you configure Tableau Server for local authentication, you cannot set password policies or account lockout on failed password attempts. If you require these account safeguards, then you should use Active Directory authentication.

Active Directory

If Tableau Server is configured to use Active Directory authentication, then credentials are managed and verified by Active Directory. 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 local identity store.

In this scenario, Tableau Server must be installed in a domain in Active Directory. Tableau Server will sync user and group metadata from Active Directory to the identity store. You do
not have to manually add users. However, after the data is synchronized, you will need to assign site and server roles. You can assign these individually, or at the group level. Tableau Server does not synchronize any data back to Active Directory. Tableau Server manages content and server access according to the site role permission data is stored in the repository.

If you are already using Active Directory to manage users in your organization, then we recommend selecting Active Directory authentication during Tableau setup to make user provisioning and management easier. For example, by synchronizing Active Directory groups, you can set minimum site role Tableau permissions for users that are synchronized in the groups. You can synchronize specific Active Directory groups, or you can synchronize them all. For more information, see Synchronize All Active Directory Groups on the Server.

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.

Data connection 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 using local authentication, while configuring Kerberos delegation, OAuth, or SAML authentication to specific data sources. See Data Connection Authentication.

Single sign-on options for Tableau Server

Tableau Server supports several single sign-on (SSO) solutions. With SSO, users don't have to sign in to Tableau Server explicitly. Instead, the credentials they've used to authenticate already (for example, by signing in to your corporate network) are used to authenticate them to Tableau Server, and they can skip the step of entering a user name and password. With SSO, the user's identity as established externally is mapped to a user identity defined in the Tableau Server identity store.
When you configure Tableau Server to use an SSO solution, all authentication is handled by the SSO solution. However, Tableau Server will manage user access to Tableau resources based on the site roles stored in the identity store and permissions set on those resources.

Tableau Server supports these types of SSO:

- **SAML**. You can configure Tableau Server to use SAML (security assertion markup language) for SSO. 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.

  You can use SAML to access Tableau Server if you have configured Active Directory or local authentication on Tableau Server. For more information, see SAML.

- **Kerberos**. If Kerberos is enabled in your environment and if the server is configured to use Active Directory authentication, you can provide users with access to Tableau Server based on their Windows identities. You cannot use Kerberos if your Tableau Server is configured for local authentication. For more information, see Kerberos.

- **OpenID**. 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 on Tableau Server, the server must be configured to use local authentication. Active Directory authentication is not supported. For more information, see OpenID Connect.

- **Trusted Authentication**. Trusted authentication 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.
• **Integrated Windows Authentication.** If you have configured Tableau Server with Active Directory authentication, you can enable automatic logon. Automatic logon uses Microsoft SSPI to sign in your users based on their Windows user name and password. Users are not prompted for credentials, which creates an experience similar to single sign-on (SSO). To enable automatic logon see, Configure General Server Options.

Related topics

• Trusted Authentication

• REST API: Signing In and Out (Authentication)

**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 are permission sets that are assigned to a user, such as System Administrator, Publisher, or Viewer. The site roles define collections of capabilities (delete, save, view, and others) that can be granted to users or groups on Tableau Server.

Site roles define who is an administrator. Administrators can be assigned at the site or server level. Site roles also determine whether non-admin users are allowed to publish to
the server from Tableau Desktop. In general, site roles determine the maximum capabilities that can be granted for each non-admin user. For example, if a user’s site role is Interactor, the user cannot publish to the server, no matter what other permissions the user has, because the Interactor role denies permission to publish.

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 resource.

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 resources 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 resource-based.** Permissions are assigned to individual resources and are granted to users or groups. Permissions are evaluated for projects, workbooks, views, and data sources.

- **Permissions are implicitly denied, and non-admin users must explicitly be allowed to access resources.** The process by which Tableau Server determines the “allow” or “deny” permission is explained in detail in the topic, How Permissions are Evaluated.

- **Permissions inheritance exists only in locked projects and in workbooks with tabbed views.** When content permissions are locked to the project, its workbooks, views, and data sources will always use the default permissions in the project. In the case of workbooks saved with the option **Show sheets as tabs**, views will use the workbook permissions. For more information, see Content Permissions and Ownership.

- **In a project that is not locked, initial permissions are a one-time copy of the container item’s permissions.** A workbook, view, or data source will start with the default permissions, but authorized users can subsequently edit permissions on those
resources. For more information on default permissions and projects, see Set Project Default Permissions and Lock the Project.

Tableau Server provides a flexible permissions infrastructure that allows you to manage access to all content for countless scenarios. See Content Permissions and Ownership for more detailed information.

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 may require authorization that is outside the scope of Tableau Server’s authority. If users publish an external data source, then Tableau Server will manage access and capabilities of data source. But if users embed an external data source in a workbook, then it’s up to the users who publishes the workbook to determine how other users who open the workbook will authenticate with the data source.
- 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 may 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.

Server Administration: Authorization for Configuring Tableau Server

One or more users must have Windows local admin permissions to configure Tableau Server and to run `tabadmin` configuration options commands.
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>Is database security possible per Tableau Server user?</td>
</tr>
<tr>
<td>Authentication mode</td>
<td></td>
</tr>
<tr>
<td><em>Window NT Integrated Security (Windows Authentication)</em></td>
<td>Server Run As account</td>
</tr>
<tr>
<td></td>
<td>Impersonate via server Run As account</td>
</tr>
</tbody>
</table>

- *Note: The ‘No*’ indicates that this mode is not recommended for security reasons.*
<table>
<thead>
<tr>
<th><strong>Database Connection Options</strong></th>
<th><strong>Data Security Questions</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Database login account uses...</td>
<td>Is database security possible per Tableau Server user?</td>
</tr>
<tr>
<td><strong>Authentication mode</strong></td>
<td>Yes</td>
</tr>
</tbody>
</table>

**Username and Password**

| **Prompt user:** Viewers are prompted for their database credentials when they click a view. Credentials can be saved. | Yes | No | No |

| **Embedded credentials:** The workbook or data source publisher can embed their database credentials. | No | Yes | Yes |

| **Impersonate via embedded password:** Database credentials with impersonate permission are embedded. | Yes | No* | No |

* 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.

Some of the options described above require configuration steps that must happen during Tableau Server Setup or before you publish a workbook or data source. See the following topics for more information:

- Server Settings (General)
- Enable Kerberos Delegation
- OAuth Connections
- Run As User
- SQL Server Impersonation
- **User Filters and Data Source Filters** in the Tableau Help.

Related Topics

Regenerate a Password for the Tableau Server PostgreSQL Database (Repository)

Manage Server Secrets

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.
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 External SSL.

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 clickjack 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.

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 server components. By default, SSL is disabled for communications between server components and the repository.
For more information, see Configure Internal SSL.

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.
Run As User

The *Run As User* is a Windows account that Tableau Server uses ("runs as") when it accesses resources. For example, Tableau Server reads and writes files on the computer where Tableau Server is installed. From the perspective of Windows, Tableau Server is doing this as the Run As User. In some cases, Tableau Server may use the Run As User account to access data from external sources, such as databases or files on a shared network directory.

As you plan your Tableau Server deployment, you need to determine if the default Run As User, configured to run under the context of the local Network Service account (NT Authority\Network Service), will suffice for your needs. If it does not, then you will need to update the Run As User to run under a domain account that has access to the resources in your Active Directory domain(s).

In either case, it’s important to understand the security implications of the account that Tableau Server uses for the Run As User. Specifically, if Tableau Server needs to access other servers, file shares, or databases that use Windows authentication, then the account that is configured for Run As User will be used to access those resources. The account that is configured for Run As User must also have elevated permissions to the local Tableau Server. A general best security practice is to limit the scope of all user accounts to the minimum required permissions. We make the same recommendation to you as you plan Run As User.

You set or update the Run As User account in the Tableau Server Configuration utility. The utility sets permissions for the Run As User, but if you are unsure if the account you want to use for Run As User satisfies the requirements, or if you have changed the Run As User and are getting permission errors, see Required Run As User Account Settings.
Default Run As User account: Network Service

The Network Service account is a predefined local account with limited permissions that exists on all Windows computers. While it has limited administrative access to the local computer on which it runs, it does have more access to resources than members of the Active Directory default Users group. For example the Network Service group can write to the registry, the event log, and has special rights to log on for application services.

By default, the Run As User is set to a local account called Network Service. Use the default Network Service account when:

- You are using local authentication for Tableau Server.
- All users in your organization include extracted data in the workbooks that they are uploading to Tableau Server.
- You are running Tableau Server in a single-server deployment.
- External data sources that your users access through Tableau Server do not require Windows NT integrated security or Kerberos. In most data-access scenarios, Microsoft SQL Server, MSAS, Teradata, and Oracle databases require Windows NT integrated security.

While the Network Service account can be used to access resources on remote computers within the same Active Directory domain we do not recommend using the default account for such scenarios. Instead, configure a domain account for Run As User if Tableau Server must connect to data sources in your environment. See Create and Update the Run As User Account.
Run As User account: Domain user

For all Active Directory scenarios, we recommend updating the Tableau Server Run As User with a domain user account. Update the Run As User to a domain user account when data sources accessed through Tableau Server require Windows NT integrated security or Kerberos.

If you have deployed a distributed deployment of Tableau Server, then you can update the Run As User account with either a domain user or a Windows workgroup user. In either case, you must use the same user account for all server nodes. See Distributed Requirements for more information.

To configure your environment to use a domain account, see Create and Update the Run As User Account.

Configure Data Source Connection Settings

To automatically authenticate your users when the workbook they’re accessing connects to a live, NT-authenticated data source, configure your Tableau data connection with the **Use Windows NT Integrated security** option selected:

- **Windows NT Integrated Security**: Authenticated with the server’s Run As User account
- **Username and Password**: Each Tableau Server user is prompted for database credentials
Create and Update the Run As User Account

If you are operating in an environment where a majority of your data sources are authenticated in the context of Active Directory (Windows NT integrated security) then you will need to configure the Run As User to use a domain account, not the local account (NetworkService) that’s the default.

There are two steps:

1. Create the Run As User account in Active Directory
2. Update Tableau Server to use the Run As User account

Creating the Run As User account

Follow these best practices:

- Create a dedicated account in Active Directory for the Tableau Server Run As user account. In other words, don’t use an existing account. By using a dedicated account you can be sure that the data resources that you permission for Tableau Server are only accessible by Tableau Server Run As User.
- Do not use an account with any kind of domain administrative permissions. Specifically, when you create an account in Active Directory, create an account in the domain User Group. Do not add the account that you create to any Active Directory security groups that needlessly elevate the permissions for the account.
- Permission the data sources in your directory for this one account. The account that you’ll use for Run As User only needs Read access to the appropriate data sources and network shares.
- If users in your organization authenticate with smart cards, disable the smart card logon option for the Run As User account.

Updating the Run As User account in Tableau Server

After you have created the Run As User account in Active Directory, configure Tableau Server to use that account as the Run As User. See Configure General Server Options for information on how to update the Run As User account. After you update the Run As User, Tableau Server (tabconfig) will automatically configure permissions on the local computer for the Run As User account that you have entered.
If you have changed the Run As User account, then we recommend revoking the permissions for the previous account. See Revoke Run As User Account Permissions.

If you have installed Tableau Server on a drive other than the system drive, then you will need to configure the system drive to allow the Run As User additional permissions. The system drive is the drive where Windows is installed. For example, if you have installed Windows on the C:/ drive, then C:/ is your system drive. If you install Tableau Server on any other drive (D:/, E:/, etc), then you will need to configure permissions for the Run As User account on the system drive. See Required Run As User Account Settings for more information.

Reapplying Permissions

To reapply permissions to your Run As User account, you must first reset permissions by applying them to the default NetworkService account. This procedure requires you to restart Tableau Server services twice, so run this procedure during off hours.

1. Open the Tableau Configuration Utility and click the General tab.

2. In Server Run As User section, enter NT AUTHORITY\NetworkService in the User field. The NetworkService account is a predefined Windows account that is used as the default account when Tableau Server is installed. This account does not require a password.

3. Click OK to reset permissions to the NetworkService account. This operation typically takes about 5 minutes, but may take longer. When finished, a tabconfig notification will be displayed: Settings have been saved. Changes will take effect when Tableau Server next starts. Click OK.

4. Use tabadmin or the Tableau Server utilities to restart the server.

5. After Tableau Server services start up again, follow the previous four steps, using the original Run As User account in Step 2.
Related tasks

The Run As User is central to many operations on Tableau Server, especially those that are involved with remote data access. To avoid access errors, review the tasks here and follow the links for those that apply to your scenario.

- If you are running Tableau Server in an organization with multiple Active Directory domains, see Domain Trust Requirements.
- Enabling Kerberos single sign-on requires additional configuration related to the Run As User. To enable Kerberos single sign-on with Tableau Server, see Kerberos.
- Enabling impersonation requires additional configuration related to Run As User. To deploy and enable impersonation with Microsoft SQL Server, see Impersonate with Embedded SQL Credentials.
- If you have installed Tableau Server onto the non-system drive, then you will need to manually set some permissions for the Run As User. See Required Run As User Account Settings for more information.
- If you have changed the Run As User account, then we recommend revoking the permissions for the previous account. See Revoke Run As User Account Permissions.
- If your organization uses a forward proxy solution, then you may need to reconfigure the local LAN settings on the Tableau Server with the Run As User account. See Configure a forward proxy server for more information.

Troubleshoot Run As User

As discussed in the topic, Run As User, Tableau Server requires administrative-like access to the machine on which it is installed. Therefore, when you update the Run As User in Tableau Server Configuration, a background process will configure permissions on the Tableau computer for that account. However, in some complex deployment scenarios you may need to verify or manually configuration the Run As User permissions on the local Tableau Server computer. Use this section to verify how permissions are configured on the machines running Tableau Server in your deployment. This section also includes procedures that describe how to set permissions and configure security policies for the Run As User.
Required Run As User Account Settings

The Run As User account needs permissions that allows it to modify files and registry settings. In addition, because the Run As User is used as the security context for the Tableau Server Application Manager service (tabsvc), the account must also be given rights to log on as a service.

These permissions are set automatically when you update the Run As User account in Tableau Server Configuration as described in the topic, Create and Update the Run As User Account.

If you have changed the Run As User account, then we recommend revoking the permissions for the previous account. See Revoke Run As User Account Permissions.

If you have recently changed Run As User or are getting permission errors, use this section to confirm that Tableau Server meets the permission requirements that are detailed here. If you’re running a distributed installation, all Run As User permission configurations must be the same across the primary server and all worker nodes.

**Note:** Do not hide the files created by the Tableau Server installer.

Verify Folder Permissions

The account the Tableau Server service runs under is referred to as the Run As User account. The Run As User account needs permission to specific folder resources on the Windows computer.

This topic provides:

- An accounting of the permissions that are required by the Run As User
- Information about installing Tableau Server in non-default locations.
- How to reapply permissions on an existing Run As User account using the Tableau Configuration Utility
This topic refers to the drive where Windows is installed as the system drive. The system drive is equivalent to the Windows environmental variable, %WINDIR%. The drive where Tableau Server is installed is referred to as the install drive.

<table>
<thead>
<tr>
<th>Resource</th>
<th>System or install drive</th>
<th>File Path</th>
<th>Permissions required</th>
</tr>
</thead>
<tbody>
<tr>
<td>folder</td>
<td>system</td>
<td>SYSTEMROOT:\windows\system32</td>
<td>Read, List folder contents</td>
</tr>
<tr>
<td>executable</td>
<td>system</td>
<td>SYSTEMROOT:\windows\system32\cmd.exe</td>
<td>Read &amp; execute, List folder contents</td>
</tr>
<tr>
<td>Drive root</td>
<td>install</td>
<td>root, for example,Local Disk (C:)</td>
<td>Read, List folder contents</td>
</tr>
<tr>
<td>folder</td>
<td>install</td>
<td>PROGRAMFILES\Tableau\Tableau Server</td>
<td>Modify</td>
</tr>
<tr>
<td>folder</td>
<td>install (on system drive)</td>
<td>\ProgramData\Tableau\Tableau Server\</td>
<td>Modify</td>
</tr>
<tr>
<td>folder</td>
<td>install (on non-system drive)</td>
<td>\Tableau\Tableau Server\data\</td>
<td>Modify</td>
</tr>
</tbody>
</table>
When you update the Run As User account in the Tableau Server Configuration utility, a background process (tabconfig) will configure the folder permissions on the Tableau computer for the Run As User account that you specify.

In this case, where you are installing on the system drive into the default folder (C:\Program Files\Tableau), the configuration of folder permissions will be handled by the tabconfig process when you update the Run As User account in Tableau Server Configuration. You do not need to verify or change any folder permissions for this scenario. If you install Tableau Server onto a different drive, you will need to manually configure some permissions.

Installing on non-system drive or in a different folder

If you have installed Tableau Server on a drive other than the system drive, then you will need to configure the system drive to allow the Run As User additional permissions. The following table describes the additional permissions that you need to configure if you install Tableau Server on a drive other than the system drive.

<table>
<thead>
<tr>
<th>Resource</th>
<th>System or install drive</th>
<th>File Path</th>
<th>Permissions required</th>
</tr>
</thead>
<tbody>
<tr>
<td>Drive root</td>
<td>install</td>
<td>root, for example, D:\</td>
<td>Read, List folder contents</td>
</tr>
<tr>
<td>folder</td>
<td>system</td>
<td>SYSTEMROOT:\windows\system32</td>
<td>Read, List folder contents</td>
</tr>
<tr>
<td>executable</td>
<td>system</td>
<td>SYSTEMROOT:\windows\system32\cmd.exe</td>
<td>Read &amp; execute, List folder contents</td>
</tr>
</tbody>
</table>
This procedure describes how to set permissions for the Run As User on a given folder in Windows. Use this procedure to set the permissions specified in the table above. This procedure shows how to modify permissions where the Windows system drive is the C:\ drive and the Tableau install drive is on D:\.

1. On the computer hosting Tableau Server (and on Tableau Worker nodes, if distributed), use Windows Explorer to right-click the Tableau Server install drive, for example Local Disk (D:), and select Properties.

2. In the Local Disk Properties Window, select the Security tab.

3. Click Edit, then Add.

4. In the Select Users, Computers, Service Accounts, or Groups dialog box, type the \<domain>\<username> for the Tableau Server Run As User account.

5. Click Check Names to resolve the account, then OK to confirm.

6. With the Tableau Server Run As User account highlighted, select List folder contents and Read.

7. On the bottom of the Security tab, click Advanced.

8. In the Advanced Security Settings for Local Disk (C:) window, click Change Permissions.

9. In the Advanced Security Settings for Tableau dialog box, highlight the Run As User account and select the Replace all child object permissions with inheritable permissions from this object check box.

10. Click OK to apply changes to all subfolders and files - this may take a few minutes.
**Note:** You may encounter one or more warning messages from Windows when you apply these changes. For example, a Windows Security warning message will appear that warns that changing permissions on the root directory may reduce security of the computer. In all cases, verify that the warnings are acceptable given the context and continue with the procedure.

11. Click **OK** to confirm changes, then click **OK** in the Local Disk (C:) Properties dialog box.

12. Follow the steps above to apply the following permissions:

   - **Read, List folder contents:** `SYSTEMROOT:\windows\system32`
   - **Read & execute, List folder contents:** `SYSTEMROOT:\windows\system32\cmd.exe`

13. Click **OK** to exit.

**Note:** In some cases, Windows will display a Recycle Bin error: "The Recycle Bin is corrupted. Do you want to empty the Recycle Bin for this drive?" Click **Yes**.

### Reapplying folder permissions

In some organizations, Group Policy or other system management solutions are used to standardize permissions and accounts on application servers. If your organization runs a such a solution, be sure to configure the system to accommodate the folder permissions required by the Run As User account. If the folder permissions for the Run As User account have been changed, you can use Tableau Configuration Utility to reapply the permissions. See Reapplying Permissions.
Verify Registry Permissions

The account the Tableau Server service runs under needs permission to modify the registry on the local machine.

In a multi-node cluster, the registry permissions are only granted on the primary node in the cluster.

When you update the Run As User in Tableau Server Configuration, a background process (tabconfig) will configure the registry permissions on the Tableau computer for the account you specify. It’s unlikely that you will need to apply these permissions manually.

Verify that the Run As User has been granted permissions to the following registry branches. If the account that you have specified as the Run As User is a member of the local administrative group or a member of the Domain Admins security group, then the account will not be displayed on the Permissions page.

- HKEY_CURRENT_USER\Software\Tableau
- HKEY_LOCAL_MACHINE\Software\Tableau

Permissions

Tabconfig will grant Read permission and the following Special permissions to these branches:

- Query Value
- Set Value
- Create Subkey
- Enumerate Subkeys
- Notify
- Write DAC
- Write Owner
- Read Control

To view or edit permissions on registry directories:
1. Open the Registry Editor by entering `regedit` in Windows Run, and then clicking OK.
2. In Registry Editor, navigate to the directory where you want to view or edit permissions. Right-click the directory, and then click Permissions....
3. In Permissions, on the Security tab, select the Run As User account, and then click Advanced.
   If you are adding your Run As User account, then click Add and follow the Windows process for adding a user account to the Security tab. After you have added the account, then select the Run As User account, and then click Advanced.
4. In Advanced Security Settings, on the Permissions tab, select the Run As User account, and then click Edit.
5. On the Permission Entry, under Basic permissions, verify that Read and Special permissions are selected. Verify that Only apply these permissions to objects and/or containers within this container is not selected.
6. To view or edit Special permissions, click Show advanced permissions.
7. Under Advanced permissions, verify that the permissions enumerated at the beginning of this topic are selected. Verify that Only apply these permissions to objects and/or containers within this container is not selected.
8. If you have set new permissions, then click OK through the multiple windows to finish.
   If you have viewed permissions and not edited anything, then click Cancel to close all windows.

Verify the Local Security Policy

After you specify a Run As User account in Tableau Server Configuration (as described in the topic, Create and Update the Run As User Account), a background process (tabconfig) will update the local security policy on the computer running Tableau Server. Tabadmin will update the local security policy to give "log on as a service" permissions to the Run As User account. This elevated policy is required because the Run As User is used as the security context for the Tableau Server Application Manager service (tabsvc).

**Note:** If the Run As User account that you specify in Tableau Server Configuration is a member of the local administrators or a domain administrator, then tabadmin may not update the local security policy. Updating the Run As User with an account that is a
In some cases, you may need to manually set security policy for your Run As User. For example, some organizations run Windows Group Policy that remove "Log on as a service" rights that have been set on user accounts. Or an organization may run a policy that creates a permission conflict by specifying "Deny log on as a service." If your organization does this, then you will need to disable or edit such Group Policies so that your Run As User account is not affected.

The following procedure describes how to configure security policy, **Log on as a service**, manually. You can also use the procedure below to verify that your Run As User is appropriately configured with local security policy rights. For example, you should verify that the Run As User account is not specified on the **Deny log on as a service** policy.

If you are running a distributed installation, then configuration must be the same across the primary and all worker nodes.

To verify or update the local security policy:

1. Select **Start > Control Panel > Administrative Tools > Local Security Policy**.

2. In Local Security Policy, open **Local Policies**, select **User Rights Assignments**.

   To verify or set Log on as a service policy:
a. Right-click **Log on as a service** policy and then click **Properties**.

![Local Security Policy](image)

b. In **Log on as a service Properties**, click **Add User or Group**.

c. Type the `<domain>\<username>` for the Tableau Server Run As User account (for example: MYCO\tableau_server), and click **Check Names**.

d. When the account resolves correctly, it is underlined. Click **OK**.

To verify Run As User account is not specified in the Deny log on as a service policy:

a. Right-click **Deny log on as a service** policy, and then click **Properties**.

b. In **Deny log on as a service Properties**, verify that the Run As User account is not listed. If it is, remove it. When you are finished, click **OK**.

3. Click **OK** to close the Local Security Settings windows.
Verify Tableau Service Settings

Confirm that Tableau services are assigned the correct Log On and Startup values. If you are running a distributed installation of Tableau Server, perform these steps on the workers as well as on the primary.

1. Log on as administrator to the computer running Tableau Server.

2. On the Tableau Server computer, select **Start > Control Panel > Administrative Tools > Computer Management > Services and Applications > Services.**

3. Open Services and Applications, then click **Services.** Confirm that the following services have the correct settings:

<table>
<thead>
<tr>
<th>Service Name</th>
<th>Logon Value</th>
<th>Startup Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>FLEXnet Licensing Service</td>
<td>Local System</td>
<td>Automatic</td>
</tr>
<tr>
<td>Secondary Logon</td>
<td>Local System</td>
<td>Automatic</td>
</tr>
<tr>
<td>Tableau Server Application Manager (tabsvc)</td>
<td><code>&lt;domain&gt;\&lt;username&gt;</code> This is the Run As User account. See below. If you have not specified a Run As User account, then Network Service account is used.</td>
<td>Automatic</td>
</tr>
<tr>
<td>Tableau Server License Manager (tablicsrv) (Runs on primary node only)</td>
<td>Local Service</td>
<td>Automatic</td>
</tr>
<tr>
<td></td>
<td>The License Manager relies on default Windows folder permissions that are applied to the Local Service. If you are seeing licensing errors in the tabadmin log files, then you may need to modify permissions on the Tableau installation directory.</td>
<td></td>
</tr>
<tr>
<td>Service Name</td>
<td>Logon Value</td>
<td>Startup Value</td>
</tr>
<tr>
<td>--------------</td>
<td>-------------</td>
<td>---------------</td>
</tr>
<tr>
<td></td>
<td>See Verify Folder Permissions for more information.</td>
<td></td>
</tr>
</tbody>
</table>

**Note:** Do not change the default settings on the **Recovery** tab of the **Tableau Server Application Manager Properties** dialog box; leave the settings for failure recovery as **Take No Action**. If you change these settings, Tableau Server will restart after being stopped via the **tabadmin** command or **Stop Tableau Server** command.

Changing the Log On Value

To change the **Log On** value for Tableau Server (tabsvc) to the Run As User account:

1. Select **Start > All Programs > Tableau Server > Stop Tableau Server**.
2. Select **Start > All Programs > Tableau Server > Configure Tableau Server**.
3. On the General tab, enter the domain, user name, and password for Tableau Server’s Run As User account.
4. Click **OK**, and then select **Start > All Programs > Tableau Server > Start Tableau Server**.

Revoke Run As User Account Permissions

Changing the Run As User account in Tableau Server Configuration Utility does not remove the permissions from the previous account. Therefore, after you change the Run As User account, we recommend manually revoking permissions from the previous account as a secure best practice. Follow the procedure below to revoke permissions from the previous Run As User account.
If you have changed your Run As User account and your organization uses a forward proxy solution, then you may need to reconfigure the local LAN settings with the new Run As User account. See Configure a forward proxy server for more information.

You must be logged onto the Tableau Server computer with an administrator account to perform the following procedures.

Remove folder permissions

Tableau Server configures permissions on Windows folders according to the location where you install Tableau Server. Before you remove folder permissions, review Verify Folder Permissions to identify the resources for which you will need to remove permissions. Run the following procedure for each of the resources you have identified:

1. For each resource (drive, folder, executable), right-click the resource, and then click Properties.
2. On the resource property page, click the Security tab, and then click Edit to change permissions.
3. On the Permissions page, select the previous Run As User account and then click Remove.
4. Click OK.

Remove registry permissions

Remove the previous Run As User account from the following registry locations:

- HKEY_CURRENT_USER\Software\Tableau
- HKEY_LOCAL_MACHINE\Software\Tableau

Warning: Editing the Windows registry incorrectly can have harmful effects on your computer.

1. Open the Registry Editor by entering regedit in Windows Run, and then clicking OK.
2. For each registry directory, right-click the Tableau folder, and then click Permissions.
3. In the **Permissions for Tableau** page, select the previous Run As User account, and then click **Remove**.

4. Click **OK**.

**Remove security policies**

Remove the previous Run As User account from the following security policies:

- “Log on as a service” policy
- “Allow log on locally” policy

1. Select **Start > Control Panel > Administrative Tools > Local Security Policy**.
2. In **Local Security Policy**, open **Local Policies**, select **User Rights Assignments**.
3. For each policy:
   a. Right-click the policy and then select **Properties**.
   b. On the policy property page, select the previous Run As User account, and then click **Remove**.
   c. Click **OK**.

**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.
**Important:** We do not recommend installing Tableau Server on a computer that is running IIS. Additionally, if you are running antivirus software, you should follow the recommendations in the Knowledge Base to exclude the Tableau Server directories. The procedures in this chapter assume that you've installed Tableau Server onto a clean computer.

In this article:

- How Tableau communicates with the internet
- Configure a forward proxy server
- Configure a reverse proxy server

### 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 maps.tableausoftware.com using port 443. If it cannot make this connection, maps may fail to load.

- Licensing. Tableau products connect to the internet to activate license keys. Unless you activate Tableau software with the Offline Activation Tool, all Tableau products must have continuous access to the internet to validate their licenses.

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

  - licensing.tableau.com:443
  - o.ss2.us
  - ocsp.rootg2.amazontrust.com
- ocp.sca0a.amazontrust.com
- ocp.sca1a.amazontrust.com
- ocp.sca1b.amazontrust.com
- ocp.sca2a.amazontrust.com
- ocp.sca3a.amazontrust.com
- ocp.sca4a.amazontrust.com
- crt.sca0a.amazontrust.com
- crt.sca1a.amazontrust.com
- crt.sca1b.amazontrust.com
- crt.sca2a.amazontrust.com
- crt.sca3a.amazontrust.com
- crt.sca4a.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 can run without internet access, but in most organizations, the scenarios in the list require Tableau to be able to access the internet.

To configure access to the internet from Tableau Server, you should use a forward proxy.
Note: Both Tableau Desktop and Tableau Server need to communicate with the internet for mapping, licensing, and external data. In this article, we focus on Tableau Server, which has specific requirements for configuring internet access. Do not set up Tableau Server on the computer that's acting as your organization's internet gateway.

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.

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.

Note: If you know that none of your users need access to map data or online data sources in the workbooks that they'll be publishing to Tableau Server, and if you are configuring Tableau Server for offline licensing, you can skip this section. Otherwise, you'll need to configure Tableau Server to connect to the internet.

Configuring Tableau Server to work with a forward proxy

The steps for configuring internet options on the Tableau Server computer depend on which of these scenarios describes your enterprise:
• **Your organization doesn’t use a forward proxy solution.** If your organization is not running a proxy solution and the computer where you are installing Tableau Server can communicate with the internet, you don’t need to follow the procedures here.

• **A proxy solution is deployed, and automatic configuration files define connection settings.** If your organization uses automatic configuration files (such as PAC or .ins files) to specify internet connection information, you can use this information in the Local Area Network (LAN) Settings dialog box in Windows. For more information, see [Enable Automatic Detection and Configuration of Browser Settings](#) on the Microsoft support site.

• **A proxy solution is deployed, but automatic configuration files are not deployed.** For this scenario, you must configure LAN settings on the Windows computer that is running Tableau Server so that connections to your proxy server are run under the security context of the Run As User account. You must also configure `localhost` and other internal Tableau Server instances as exceptions.

The following procedure describes the steps for the last scenario—a proxy solution without automatic configuration files, where Tableau Server is running on Windows Server.

**Note:** If you are using a distributed installation of Tableau Server, perform the following procedures on the primary server and on each worker node.

**Step 1: Add the Run As User account to the Local Administrators group**

To perform this procedure, you must log onto the Tableau Server computer as the Run As User. By default, the "log on locally" policy is not applied to the Run As User account. Therefore, you must temporarily add the Run As User account to the Local Administrators group.

If you haven’t installed Tableau Server on the computer yet, see [Run As User](#) for more information about creating the Run As User account. If you already installed Tableau Server and set the Run As User setting, you can determine the Run As User account name by
logging onto Tableau Server. The Tableau Server Run As User is listed on the **General** tab of the **Tableau Server Configuration** window. To access the configuration utility, in the Windows Start menu, search for **Configure Tableau Server**.

Add the Run As User to the Local Administrators group using steps in *Add a member to a local group* on the Microsoft website. When you've finished configuring the forward proxy information, you'll remove the Run As User account from the Local Administrators group.

**Step 2: Configure the proxy server in Windows LAN Settings**

1. **Using the Run As User account, log onto the computer where Tableau Server is installed or will be installed.**

2. **Open the **Local Area Network (LAN) Settings** dialog box.** (A quick way to get to this dialog box is to search for **Internet Options** in the Windows Start menu. In the **Internet Properties** dialog box, click the **Connections** tab, and then click **LAN settings**.)

3. **Under **Proxy server**, select **Use a proxy server for your LAN**, enter the proxy server address and port, and then select **Bypass proxy server for local addresses**.**
Leave this dialog box open and continue to the next step.

Step 3: Add exceptions to bypass the proxy server

You 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.

1. In the LAN settings dialog box, click **Advanced**. (This button is available only if you’ve selected the option to use a proxy server for your LAN.)

2. In the **Proxy Settings** dialog box, enter **localhost** in the **Exceptions** field. In addition, enter the server names and IP addresses of other Tableau Server computers in the same cluster. Use semicolons to separate items.

3. Close the proxy settings dialog box and the Local Area Network (LAN) Settings dialog box.
box.

4. In the **Internet Properties** dialog box, click **OK** to apply the settings.

Stay logged onto the computer and continue to the next step.

**Step 4: Verify proxy settings are not configured as environment variables**

Some organizations configure forward proxy settings as environment variables in the Windows operating system. If such settings are configured on the computer that is running Tableau Server, then you must verify that they do not conflict with the configurations you’ve completed here.

1. On the computer that is running Tableau Server, enter "advanced system settings" in the search box and then click Enter to open the **System Properties** box.

2. In **System Properties**, click **Environment Variables**.

3. Scroll through the **System variables** field.

   If **http_proxy** or **https_proxy** are specified, verify that the values do not conflict with the proxy server address that you configured in the previous step.

   - If the existing settings do not conflict, then go to "Step 5: Test the proxy configuration."

   - If the existing settings do conflict, then create a new variable named **no_proxy**, and enter the host name, IP address, and port of the Tableau Server for the value. For example, localhost,192.168.0.10:80. For more information see the Microsoft MSDN article, **Set Environment Variables**.

4. Click **OK**.

**Step 5: Test the proxy configuration**

To test the new configurations, while still logged on as the Run As User on the Tableau Server computer, open a web browser and test the following Tableau mapping URL:

**Miami and Havana (blue water)**
This is the URL:

https://maps.tableausoftware.com/tile/d\mode=named\from=tableau1_2_base/-
\mode=named\from=tableau1_2_admin0-
borders\mode=named\from=tableau1_2_place_ 
labels\ol/6/17/27.png?apikey=ttab56540ba691a909b0f7d2af0f6fe7"

If the configuration is working, you see a map of Miami and Havana. This indicates that the Tableau Server computer is able to access the internet through the proxy.

Step 6: Remove the Run As User account from the Local Administrator group

After you have tested the proxy settings, remove the Run As User account from the Local Administrators group. Leaving the Run As User in the administrator group unnecessarily elevates the permissions of the Run As User group and is a security risk.

Restart Tableau Server to ensure that all changes are implemented.

Configure a reverse proxy 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.

How a reverse proxy works with Tableau Server

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 precondition 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 clients

Prior to Tableau Server 10.3, Tableau added a non-standard header (Server: Tableau) to all HTTP responses for Mobile client sessions.
Beginning with Tableau Server version 10.3, 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.

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

- Tableau Desktop and the Android version of Tableau Mobile do not support authentication with a reverse proxy. For these clients you should use a **VPN solution**, or configure your reverse proxy to route traffic from these clients 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 `tabadmin` utility. The information you need to collect corresponds to options you'll need when you run `tabadmin`.

<table>
<thead>
<tr>
<th>Item</th>
<th>Description</th>
<th>Corresponding <code>tabadmin</code> option</th>
</tr>
</thead>
<tbody>
<tr>
<td>IP address or CNAME</td>
<td>You can either enter an IP address or a CNAME for this option.</td>
<td><code>gateway.trusted</code></td>
</tr>
<tr>
<td></td>
<td>The public IP address or addresses of the proxy server. The IP address must be in IPv4 format, such as <code>203.0.113.0</code>, and it must be a static IP.</td>
<td></td>
</tr>
<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>Item</td>
<td>Description</td>
<td>Corresponding tabadmin option</td>
</tr>
<tr>
<td>----------</td>
<td>-----------------------------------------------------------------------------</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 a FQDN with information beyond the domain name, such as 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>.</td>
<td>gateway.trusted_hosts</td>
</tr>
<tr>
<td>Ports</td>
<td>Port numbers for traffic from the client to the reverse proxy server.</td>
<td>gateway.public.port</td>
</tr>
</tbody>
</table>

If you are using a distributed installation of Tableau Server, then run the following procedure on the primary node in your cluster.

1. **Open a command prompt and navigate to the Tableau Server bin directory.**
   
   1. Open a command prompt as an administrator:
2. Enter the following to change to the folder where `tabadmin.exe` is located:

```
cd "C:\Program Files\Tableau\Tableau Server\10.5\bin"
```

2. Enter the following command to stop Tableau Server:

```
tabadmin stop
```

3. 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:

```
tabadmin set gateway.public.host "name"
```

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

```
tabadmin set gateway.public.host "tableau.example.com"
```

4. 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:

```
tabadmin set gateway.trusted "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.

5. 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.

```
tabadmin set gateway.trusted_hosts "name1, name2, name3"
```

For example:

```
tabadmin set gateway.trusted_hosts "proxy1.example.com, proxy1, ftp.example.com, www.example.com"
```

6. 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:

```
tabadmin set gateway.public.port "443"
```

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

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

```
tabadmin config
```

8. Enter the following command to restart the server:

```
tabadmin start
```
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:

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

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><code>REMOTE_ADDR</code></td>
<td>Tableau Server needs these headers</td>
<td>The IP address that you set</td>
</tr>
<tr>
<td><code>XFF</code></td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>HOST</code></td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>XFH</code></td>
<td></td>
<td></td>
</tr>
<tr>
<td><code>XFP</code></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Header Name</td>
<td>Description</td>
<td>Example</td>
</tr>
<tr>
<td>-------------</td>
<td>-------------</td>
<td>---------</td>
</tr>
<tr>
<td>X-FORWARDED-FOR (XFF)</td>
<td>Used 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>ingateway.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>
<tr>
<td>HOST and X-FORWARDED HOST (XFH)</td>
<td>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.</td>
<td>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.</td>
</tr>
<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</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</td>
</tr>
</tbody>
</table>
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.

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.</td>
<td>Sign in to Tableau Server or Online</td>
</tr>
<tr>
<td>Log into Tableau Server from a web browser.</td>
<td>Sign in</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>Online</td>
<td>Download workbook from a web browser.</td>
</tr>
<tr>
<td>--------</td>
<td>-------------------------------------</td>
</tr>
<tr>
<td>tabcmd</td>
<td>Check to make sure tabcmd (from a non-server client) works.</td>
</tr>
</tbody>
</table>

**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 in the Tableau Server Configuration utility.

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 the Run As User 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 Run As User account that is used for Tableau Server can access the data source.

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.

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\jsmith`, where `example` is the NetBIOS name and `jsmith` 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 users individually

In most cases, Tableau Server uses the `sAMAccountName` value for the user name. When you import users individually from Active Directory (either by typing in their names or by
using a CSV file), Tableau queries Active Directory with the user name that you provide. If a match is found, then that name is imported into Tableau Server and it becomes the name that the user enters in order to sign in to Tableau Server.

The user name that Tableau Server will import into the identity store will be the `sAMAccountName` value unless one of the following is true:

- If the user name that you specify is longer than 20 characters.
- If the user name that you specify contains an @ character.

If the user name you enter meets either of the these conditions, then Tableau will import the UPN prefix of the `userPrincipalName` attribute, which will become the user's Tableau logon user name.

If user names were inadvertently imported using UPN names, you can delete the accounts in Tableau Server and then reimport those accounts using the `sAMAccountName` value for the user name, as shown in **User logon name (pre-Windows 2000)** in the Windows **Active Directory Users and Computers** administrative console.

**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.
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 `j smith` in that domain could log on to Windows using either of the following user names:

- `west.na.corp.example.com\j smith`
- `SEATTLE\j smith`

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.

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** 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.

- **Minimum recommendations** are higher than 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

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. For more information on how the Setup program determines hardware, see "Determining Computer Hardware," below.

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>Minimum Hardware Requirements</th>
<th>CPU</th>
<th>RAM</th>
<th>Free Disk Space</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>2-core</td>
<td>8 GB</td>
<td>15 GB</td>
</tr>
</tbody>
</table>

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 the various factors like whether or not you will be using extracts. For more information, see Disk Space Requirements.

- 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.
Note: For Tableau Server 10.5, you need a minimum of 2 physical cores. If you are installing on an Amazon EC2 instance, this means 4 vCPUs. For more information, see Amazon EC2 Instances.

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.

<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 or higher</td>
<td>32 GB</td>
<td>50 GB</td>
</tr>
<tr>
<td>Multi-node and enterprise deployments</td>
<td>Contact Tableau for technical guidance.</td>
<td>Nodes must meet or exceed the minimum hardware recommendations, except nodes running backgrounder, where 4 cores may be acceptable.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Determining Computer Hardware

To determine how many physical cores a computer has, the Tableau Server setup program queries the operating system. To view hardware information that the setup program detected on your computer, open the `tabadmin.log` file in the following folder on the computer where you are installing Tableau Server:
<install directory>\ProgramData\Tableau\Tableau Server-\logs\tabadmin.log

In the tabadmin.log file, look for lines similar to the following. These lines provide information about the physical and logical cores that the setup program detected and that it used to determine the core count that is being used for licensing.

2015-04-09 14:22:29.533 -0700_DEBUG_10.36.2.32:<machine name>:_:_pid=21488_0x2cd83560__user=___request=__ Running hardware check

2015-04-09 14:22:29.713 -0700_DEBUG_10.36.2.32:<machine name>:_:_pid=21488_0x2cd83560__user=___request=__ Detected 12 cores and 34281857024 bytes of memory

2015-04-09 14:22:29.716 -0700_DEBUG_10.36.2.32:<machine name>:_:_pid=21488_0x2cd83560__user=___request=__ Hardware meets recommended specifications. Default values will be used.

Manually determining the number of cores on your computer

To determine manually how many physical cores your server has, you can use the Windows Management Instrumentation Command-line tool (WMIC). This is useful if you do not know whether your computer will meet the minimum hardware requirements for installing Tableau Server.

1. Open a command prompt.

2. Enter the following command:

   WMIC CPU Get DeviceID,NumberOfCores

   The output will display the device ID or IDs and the number of physical cores the computer has.
In this example, there are two CPUs, each with six cores, for a total of twelve physical cores. This computer would satisfy the minimum hardware requirements for installing Tableau Server.

The following command shows a longer version that lists the logical processors as well as the physical cores.

WMIC CPU Get DeviceID,NumberOfCores,NumberOfLogicalProcessors,SocketDesignation

In the above example, the server has a total of twelve physical cores, resulting in 24 logical cores.

**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.

Looking for Tableau Server on Linux? 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.

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.

In this article

Hardware Recommendations
Baseline Configurations
Beyond Baseline Configurations

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.
### Baseline Configurations

- **Single Server Installations**
- **Two Node Installations** (Specialized for extract heavy environments)

<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-core (physical), 2.0 GHz or higher</td>
<td>64 GB (8GB/physical Core)</td>
<td>500 GB - 1 TB</td>
</tr>
<tr>
<td>Multi-node and enterprise deployments</td>
<td>Nodes must meet or exceed the minimum hardware recommendations, except nodes running backgrounder, where 4 cores may be acceptable.</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**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 and the number of workbooks to Tableau Server. For more information see Disk Space Requirements.
• 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 single server node with all the processes installed on one machine.

- By default Tableau Server installer configures the number of process instances based on the hardware on the machine. We recommend that you keep the default configuration as your starting point. Below are the number of processes for an 8 core machine. These numbers are derived using the Primary Server Installation Defaults.
that Tableau Server installer uses:

- VizQL Server: Set to 2 instances (default calculation: Number of physical cores divided by 4, up to a maximum of 4).
- Backgrounder, Cache Server, and Data Server: Set to 2 instances.
- All other processes, only one instance of the process is installed, regardless of hardware.

Distributed Installations

Running Tableau Server on more than one machine is called distributed installation, or a cluster. There are various reasons why you might want to have a distributed 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 distributed environment that has at least three nodes.

Two Node Installations (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).

Server Configuration

**Tableau Server Primary Node**

- Gateway
- Search & Browse
- Application Server (VizPortal) x2
- VizQL Server x2
- Data Server x2
- Cache Server x2
- Data Engine
- File Store
- Repository (active)
- Cluster Controller
- Coordination Service

**Worker Node**

- Gateway
- Search & Browse
- Backgrounder x2
- Data Engine
- Licensing
- Cluster Controller
- Coordination Service

- On the primary node, install all the processes except for the backgrounder. Below is the number of instances of the processes for an 8 core machine. These numbers are derived using the Primary Server Installation Defaults that Tableau Server installer
uses:

- VizQL Server: Set to 2 instances. (Number of physical cores divided by 4, up to a maximum of 4).
- Cache Server, and Data Server: Set to 2 instances.
- Backgrounder: Minimum 2, maximum 4.
- All other processes, only one instance of the process is installed, regardless of hardware.
- Isolate backgrounder on the worker node. 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. When you install the backgrounder, Tableau Server automatically installs one instance of the Data Engine.

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, 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 Examples topic.

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.
- Creating a backup primary (Windows install).
- 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 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.

**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 licensed Tableau Server processes, see Tableau Server Processes.

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 AWS Cloud.

For more information on installing Tableau Server on VMware, see Tableau Server on VMware VSphere.

Beyond Baseline Configurations

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:

Tableau Server Scalability

These baseline configuration 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 10.5 Scalability whitepaper.

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
- Site roles and permissions
- Extract refresh schedules
- Steps for setting up your site

The subsequent sections go over these site components, assuming that you are familiar with
Projects

You can create projects on a site, which act as containers in which you can organize related 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 data sources and workbooks to a site must be able to sign in to it. 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. Tableau Server licenses are based on either cores or users. If the server has a user-based license, there’s an upper limit to how many users can have active accounts on the server. 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.
As we just noted, site roles act as an upper limit on permissions. Permissions determine what a user can do within the context of the site role. A user whose site role is Interactor can never publish to the site, regardless of the permissions you grant them. A user whose site role is Publisher 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.

**More Deployment Resources for Tableau Server**

**Everybody's Install Guide**

Installing Tableau Server is about as easy as it gets with server software. Still, if you're new to it, you can use someone to help you figure out what to prepare and how to go through it. And now we've got you covered.

To plan for, install, and manage a single-machine instance of Tableau Server, see the Everybody's Install Guide.

**Desktop Deployment Guide**

Tableau Desktop lets your users create workbooks and views, dashboards, and data sources and then publish this content to either Tableau Online or Tableau Server.

To install Desktop for yourself or for others, see the Tableau Desktop Deployment Guide.
Tableau Mobile Deployment Guide

Tableau Mobile lets your users stay on top of critical data—anywhere they go. This free iOS and Android app lets users explore and share content published to either Tableau Online or Tableau Server 8.2 and later.

For detailed information about app deployment, customization, and security, see the Tableau Mobile Deployment Guide.

Developer Resources

The Developer Portal on the Tableau Community is the place to go for everything related to extending and automating Tableau. There you can get access to the following:

- **JavaScript API**—Integrate Tableau views into your own web applications.

- **REST API**—Manage provisioning, permissions, and publishing on Tableau Server or Tableau Online via HTTP. The REST API gives you access to the functionality behind the data sources, projects, workbooks, site users, and sites. You can use this access to create custom applications or to script interactions with server resources.

- **Tableau SDK**—Use C, C++, Java, or Python to create extracts from any data and then to publish your extracts.

- **Web data connector**—Create a Tableau connection in JavaScript to almost any data that's accessible over HTTP. This can include internal web services, JSON data, XML data, REST APIs, and many other sources.

- **ODBC connector**—Create a connection using ODBC (Open Database Connectivity), which is a data-access protocol that's supported by a wide array of data sources. In Tableau Desktop, you can connect to any ODBC-compliant source using the built-in ODBC connector.

In addition to these resources, you can get complete documentation and examples, and collaborate with the Tableau developer community.
Go to the Developer Portal
Install and Configure

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


Run Server Setup ............................................................... 110
Activate Tableau Server .................................................... 112
Configure Tableau Server .................................................. 116
Add an Administrator Account ........................................... 155
Reconfigure the Server ..................................................... 156
Distributed Environments .................................................. 184
Automated Installation or Upgrade of Tableau Server ............. 248
Upgrade .............................................................................. 279
Navigate Server Admin Pages ............................................ 376
Move Tableau Server to Another Drive ................................. 379
Install Tableau Server in a Disconnected (Air-Gapped) Environment .................................................. 382
Install Tableau Server in the AWS Cloud ............................... 387

Run Server Setup

After you download the Tableau Server installation file, follow the instructions below to install the server.
**Important:** Do not install the 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.

1. Double-click the installation file.

2. Follow the on-screen instructions to complete Setup and install the application.

![Select Destination Location](image)

The **default installation path** is `C:\Program Files\Tableau\Tableau Server`. You can choose a different location, including a different drive, either by browsing to or typing in a new path.
3. After the installation completes, click **Next** to open the Product Key Manager window.

   If you need to support characters that are not the Latin-1 set, install the Windows Language Packs via **Control Panel > Regional and Language Options**. The language packs will need to be installed on the primary server as well as any worker machines.

### Activate Tableau Server

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 **Tableau Customer Account Center**. After installing and configuring the server, the product key manager automatically opens so you can enter your product key and register the product.

If you need to activate Tableau Server on a computer that is offline, see Activate Tableau Server Offline. If you need to activate additional product keys to add capacity to an existing Tableau Server installation, see Add Capacity to Tableau Server.

If you are activating Tableau Server as part of the install process, the Product Key Manager opens automatically. If you need to open it, in Windows, click **Start > All Programs > Tableau Server <version> > Manage Product Keys**.

**Note:** You can also find instructions for activating and registering Tableau Server on the [download help page](#).
1. Select **Activate the product**:

2. Enter or paste your product key and click **Activate**.

3. Click **Continue**.

4. Enter the fields to register Tableau and click **Register**.

5. Restart Tableau Server after registration is complete.

**Activate Tableau Server Offline**

If you are working offline you can follow the steps below to complete offline activation.
Note: The first three steps instruct you to generate a file you will use for activation. You must complete these steps using a browser other than Microsoft Edge, such as Internet Explorer. If you use Edge, the activation file is not created.

1. Open the Manage Product Keys application:
   a. On the Tableau Server computer, click Start and select All Programs > Tableau Server
   b. Right-click Manage Product Keys and select Run as administrator.

2. When the Product Key Manager opens click Activate the product.
   Paste your server product key into the corresponding text box and click Activate. You can get your product key from the Tableau Customer Portal.

3. When you are offline, online activation cannot complete, and you are prompted to save a file that you can use for offline activation. Click Save.

4. Select a location for the file and click Save. Make a note of the location where you are saving the file. You will need to copy it to a computer that has Internet access. The file is saved as offline.tlq.

5. Back in Tableau click Exit to close the Activation dialog box.

6. Copy the offline.tlq file to a computer that has Internet access, open a web browser and visit the Product Activations page on the Tableau website. Complete the instructions to submit your offline.tlq file.

   After you submit your offline.tlq file online, while your browser is still displaying the Product Activations page, a file called activation.tlf is created, and Tableau prompts you to save the file to your computer.

7. Save the activation.tlf file and move it to the computer where you are installing
Tableau Server.

8. On the computer where you are installing Tableau Server, open a command prompt as an administrator and run the following command:

   \texttt{cd "C:\Program Files\Tableau\Tableau Server\10.5\bin"}

9. Next, type \texttt{tabadmin activate --tlf <path>\activation.tlf}, where \texttt{<path>} is the location of the response file you saved from the Product Activations page.

   For example:

   \texttt{tabadmin activate --tlf <path>\activation.tlf}

   Keep the command prompt window open. The Tableau Server license is now activated unless you are directed to activate a second time.

\textit{The following steps are only necessary if you are prompted a second time to activate Tableau Server.} This happens when you are activating an offline computer that has never had an activated Tableau license on it.

9. If you are directed to activate Tableau Server again, rename your original \texttt{off-line.tlq} and \texttt{activation.tlf} files on both the Tableau Server computer and on the computer that is connected to the Internet. You will not be using these file again and renaming them will help you avoid confusing these with the new versions you will be using.

10. On Tableau Server, click \texttt{Start > All Programs > Tableau Server 10.5}

11. Right-click \texttt{Manage Product Keys} and select \texttt{Run as Administrator}.

   \textbf{Note:} This is necessary even if you are logged into the Tableau Server computer as an administrator, to avoid a potential registration error.

12. Click \texttt{Activate the product}. 

- 115 -
13. Enter your product key again (the same one you entered in step 1).

14. Save the offline.tlq file and copy it to your computer with Internet access.

15. From a computer that has Internet access, open a web browser and visit the Product Activations page again on the Tableau website. Complete the instructions.

Tableau will again create a file called activation.tlf and prompt you to save it.

16. Save the file and copy it to the computer where you are installing Tableau Server.

17. Back in the command prompt window on Tableau Server, type `tabadmin activate --tlf <path>\activation.tlf`, where `<path>` is the location of the second response file you saved from the Product Activations page.

For example:

```
tabadmin activate --tlf <path>\activation.tlf
```

Tableau Server is now activated. If you need additional assistance, contact Tableau Technical Support.

---

**Configure Tableau Server**

The Tableau Server Configuration utility opens during a Tableau Server installation. You can set configuration options at this time, as part of the installation, before the server starts. The server is started at the end of the installation process.

You can also run the utility after installing Tableau Server by selecting All Programs > Tableau Server 10.5 > Configure Tableau Server on the Windows Start menu. You need to stop the server before making any configuration changes. See Reconfigure the Server for steps.

There are two things to keep in mind about the settings you specify in the Configuration dialog box:
• **Settings are system-wide**: The settings you enter apply to the entire server. If the server is running multiple sites, these settings affect every site.

• **User Authentication is "permanent"**: The User Authentication setting (on the General tab) can only be set when you are installing Tableau Server for the first time. You can change all of the other settings after installation by stopping the server and rerunning the Configuration utility.

See the topics to the right for details on the different Configuration tabs.

**Configure General Server Options**

Use the following sections to help you configure options on the General tab:

• Server Run As User

• User Authentication

• Gateway

• Firewall

• Sample data

• Upload Crash Data

**Server Run As User**

By default, Tableau Server runs under the Network Service account. To use an account that will accommodate NT authentication with data sources, specify a user name and password. The user name should include the domain name. See Run As User to learn more about using a specific user account.
User Authentication

Select whether to use **Active Directory** to authenticate users on the server. Select **Use Local Authentication** to create users and assign passwords using Tableau Server’s built-in user management system. You cannot switch between Active Directory and Local Authentication later.

Tableau Server supports several types of SSO solutions: OpenID, SAML, and Kerberos. It’s important to understand how the decision about whether to use Active Directory or local authentication affects SSO:

- OpenID requires local authentication.
- Kerberos requires Active Directory authentication.
- SAML works with either Active Directory or local authentication. However, if you plan to configure Tableau Server for site-specific SAML authentication, you must select local authentication.

If you use Active Directory:

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 select **Enable automatic logon** if you plan to configure Tableau Server for **SAML, trusted authentication**, or for a **proxy server**.

Be sure to type the fully qualified domain name (FQDN) and nickname (NetBIOS name).

**To determine the FQDN:** Select **Start > Run** then type `sysdm.cpl` in the Run textbox. In the System Properties dialog box, select the **Computer Name** tab. The FQDN is shown near the middle of the dialog box. The first time your users sign in, they will need to use the
fully qualified domain name (for example, myco.lan\jsmith). On subsequent sign-ins, they can use the nickname (NetBIOS name), for example, myco\jsmith.

Gateway

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.

Firewall

Select whether to open a port in Windows firewall. If you do not open this port, users on other machines may not be able to access the server.

Sample data

Select whether to include sample data and users. The Include sample data and users option installs several sample workbooks and data, which can help you get familiar with Tableau Server (especially if you are installing a trial version of the product). If you select Include sample data and users, the first user created in Tableau Server will be assigned as the owner of sample workbooks and data. To change the assigned owner, see Manage Content Ownership.
Server Crash Reporter

Select whether to allow Tableau Server to upload crash data. If you select the **Enable Crash Uploading** option, Tableau Server will upload crash data if the server crashes. Crash data is packaged and encrypted, then uploaded no more than once a day, at the scheduled time. Tableau will use this data to help identify and eliminate issues that can cause crashes.

**Important:** Do not select this option if your data is subject to any privacy regulations.

For more information, see Server Crash Reporter.

You can continue to the next page to configure Caching and Initial SQL options. If you do not want to configure these options click **OK**.

Configure Data Connections

Use the options on the Data Connections tab to configure caching and specify how you want to handle initial SQL statements from data sources.

Caching

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. The Data Connections tab is where you configure aspects of caching that will apply to all data connections:
To configure caching, select from one of the following options:

- **Refresh Less Often**—Data is cached and reused whenever it is available regardless of when it was added to the cache. This option minimizes the number of queries sent to the database. Select this option when data is not changing frequently. Refreshing less often may improve performance.

- **Balanced**—Data is removed from the cache after a specified number of minutes. If the data has been added to the cache within the specified time range the cached data will be used, otherwise new data will be queried from the database.

- **Refresh More Often**—The database is queried each time the page is loaded. The data is still cached and will be reused until the user reloads the page. This option will
ensure users see the most up to date data; however, it may decrease performance.

Regardless of how caching is configured, the user can click the Refresh Data button on the toolbar to force the server to send a query and retrieve new data.

Initial SQL

When connecting to some data sources, 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. 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 in Tableau Desktop.

For performance or security reasons, some administrators may want to disable this functionality. The Data Connections tab is where you do this:
To disable initial SQL functionality, select the **Ignore initial SQL statements for all data sources** check box. Workbooks created with initial SQL statements will still open but the initial SQL commands will not be sent.

**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.

See Also:

- Server Crash Reporter

Servers

Use the options on the Servers tab to do the following:
• Adjust the number of processes running on Tableau Server,
• Configure a distributed Tableau Server environment, and
• Select the preferred active repository for failover situations.

You can also use the Server tab to add computers on which to run Tableau Server processes.

![Tableau Server Configuration](image)

### Number of processes per server

Tableau Server deployments run multiple processes. You can choose to run the processes one computer, or to distribute them across multiple computers. To improve performance, you can adjust the number of processes that run on each computer, for each process type.

For more information on changing the number of server processes for a single-server environment, see Reconfigure Processes.

For more information on how many processes to run in order to improve performance, see Performance Tuning Examples.

For more information on setting up a multi-server, or distributed, environment, see Distributed Environments.
Preferred active repository

When you configure Tableau Server after the initial installation, you have the option to specify a Preferred Active Repository. This is an optional step, and if you do not specify a preferred active repository, Tableau Server will select the active repository on startup.

Configure a preferred active repository 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

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 primary 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.

The repository is installed on the primary node by default. To move it to the worker node in a two-node installation, you need to follow manual steps using tabadmin. For more information, see Move the Repository Process in a Two-Node Installation.

Primary Server 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 primary 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>Backgrunder</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>

**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.

1. In the Tableau Server Configuration utility, click the **SMTP Setup** tab.

2. Under **SMTP Server**:
   a. Enter the name of your SMTP server.
   b. (Optional) If your account requires it, enter a user name and password for your SMTP server account.
   c. If you are not using the default SMTP port 25, change the SMTP port value.
   d. Uncheck **Enable TLS** so the connection to your mail server is unencrypted. (Encrypted SMTP connections are not supported for alerts or subscriptions.)
3. For **Send email from**, enter the email address that will send an alert if there's a system failure. The email address must have valid syntax (for example, ITalerts@bigco.com 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 **Send email from** address on a per-site basis. For more information, see What is a site.

4. For **Send email to**, enter at least one email address that will receive the alerts. If you enter multiple addresses, separate them with commas.

5. For **Tableau Server URL**, enter `http://` or `https://`, followed by the name or IP address of the Tableau server. This value will be used for the footer of subscription emails.
6. Click **OK**.

After configuring SMTP and completing the steps to configure alerts (Configure Server Alerts), a **restart of server** will trigger an email alert. This confirms that you have set up alerts correctly.

**Configure Server Alerts**

On the **Alerts and Subscriptions** tab of the Tableau Server Configuration utility, you can configure the following server alerts:

- Email alerts for system failures
- Disk space usage:
  - Recording usage history
  - Email alerts when space crosses or remains below pre-configured thresholds

In the same part of the utility, you can also enable emails for subscriptions to views. For detailed information, see Set Up a Server for Subscriptions.

**Note:** You need to configure SMTP before you can configure subscriptions or alerts. For more information, see Configure SMTP Setup.

**Alerts for system failures**

Tableau Server can send email alerts to server administrators when there is a system failure.

When you configure alerts, Tableau Server sends an email to the recipients listed in **Send email to** on the **SMTP Setup** tab any time that the data engine, repository, or gateway server processes stop or restart, or any time the primary Tableau Server stops or restarts. If you are running a single-server installation (all processes on the same machine), 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 (see Configure for Failover and Multiple
Gateways), 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.

To configure email alerts for system failures

1. On the **Alerts and Subscriptions** tab of the Tableau Server Configuration utility, select **Send email alerts for server component up, down, and failover events**.

2. Click **OK**.

**Disk space monitoring**

If Tableau Server is configured to monitor free disk space and send alerts about low disk space, when space on any node in a server installation drops below the configured thresholds, Tableau Server sends an email to the recipients listed in **Send email to** on the **SMTP Setup** tab.

**Disk space usage**

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.
To configure Tableau Server to record disk space usage

1. On the **Alerts and Subscriptions** tab of the Tableau Server Configuration utility, select **Record disk space usage information, including threshold violations**.

   ![Disk Space Monitoring](image)

2. Click **OK**.

You can configure Tableau Server to send email alerts when disk space usage on any node crosses a threshold, or remains below the threshold.

To configure email alerts for low disk space

1. On the **Alerts and Subscriptions** tab of the Tableau Server Configuration utility, select **Send alerts when unused drive space drops below thresholds**.

   ![Disk Space Monitoring](image)
2. In **Warning threshold**, enter the percentage of free disk space that Tableau Server should use a warning threshold.

   If free disk space on any node in your Tableau Server cluster drops below this percentage, Tableau Server sends a warning alert email. Alerts continue until free disk space rises above the threshold. To configure the frequency of alerts, see Step 4 below.

3. In **Critical threshold**, enter the percentage of free disk that Tableau Server should use as a critical threshold.

   If free disk space on any node in your Tableau Server cluster drops below this percentage, Tableau Server sends a critical alert email. Alerts continue until free disk space rises above the threshold. To configure the frequency of alerts, see Step 4 below.

4. In **Send email alert every**, enter the number of minutes for how often Tableau Server should send an alert.

5. Click **OK**.

**Configure External SSL**

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 web browser and the server or Tableau Desktop and the server is protected. Steps on how to configure the server for SSL are described in the topic below; however, you must first acquire a certificate from a trusted authority, and then import the certificate files into Tableau Server.

Mutual SSL authentication is not supported on Tableau Mobile.
Looking for Tableau Server on Linux? See Configure SSL for External HTTP Traffic to and from Tableau Server (Linux).

If you are running a Tableau Server cluster and you want to use SSL, see Configure SSL for a Cluster, below, for recommendations.

1. 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. Wildcard certificates, which allow you to use SSL with many host names within the same domain, are also supported.

   **Note:** Be sure to use a SHA-2 (256 or 512 bit) certificate. All major browsers will display warnings when connecting to SHA-1 certificates. By the end of 2017, it's likely that most browsers will no longer connect to servers that are presenting SHA-1 certificates.

Some browsers will require additional configuration to accept certificates from certain providers. Refer to the documentation provided by your certificate authority.

2. Place the certificate files in a folder named SSL, parallel to the Tableau Server 10.5 folder. For example:

   C:\Program Files\Tableau\Tableau Server\SSL

   This location gives the account that's running Tableau Server the necessary permissions for the files. You may need to create this folder.

3. Stop Tableau Server. From the Start menu, click **All Programs > Tableau Server 10.5 > Stop Tableau Server**.

4. Open the Tableau Server Configuration Utility by selecting **Start > All Programs > Tableau Server 10.5 > Configure Tableau Server** on the Start menu.
5. In the Configuration Tableau Server dialog box, select the **SSL** tab.

6. Select **Use SSL for server communication** and provide the location for each of the following certificate files:

   - **SSL certificate file**—Must be a valid PEM-encoded x509 certificate with the extension .crt.
   - **SSL certificate key file**—Must be a valid RSA or DSA private key file (with the extension .key by convention). If the certificate key file requires a passphrase enter it in the field, **SSL certificate key passphrase**. (The passphrase you enter will be encrypted while at rest). Alternatively, you can provide a path to a key file that is not passphrase protected.

   **Note:** If you create a certificate key file with a passphrase, you cannot reuse the SSL certificate key for SAML.

   - **SSL certificate chain file** (Optional for Tableau Server, required Tableau Desktop on the Mac)—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).

7. (optional) If you are using SSL for server communication and want to configure SSL communication between Tableau Server and clients using certificates on both the server and clients:

- Select **Use mutual SSL and automatic login with client certificates.**

  **Note:** Tableau Server does not support mutual SSL and SAML together.

- In **SSL CA certificate file**, browse to the location for the certificate file. The SSL CA certificate file must be a valid PEM-encoded x509 certificate with the extension .crt.

  **Note:** If you have multiple trusted Certificate Authorities (CAs) you can copy and paste the entire contents of each CA certificate, including the "BEGIN CERTIFICATE" and "END CERTIFICATE" lines, into a new file, then save the file as CAs.crt. In **SSL CA certificate file**, browse to the location of this new file.

8. Click **OK**.

If the Run As User account on the Tableau Server is not set to the default Network Services user, then you will be prompted for the Run As User password.

9. Start Tableau Server again. From the Start menu, click **All Programs > Tableau Server 10.5 > Start Tableau Server.**

**Configure SSL for a Cluster**

You can configure a Tableau Server cluster to use SSL. If the primary 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 earlier.

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. Place the same SSL certificate and key file that you used for the initial node on each subsequent Tableau Server node that runs a gateway process. Use the same folder location on all computers.

5. If you are using mutual SSL, place the SSL CA certificate file in the same location on
all computers that run a gateway process.

You do not need to do any additional configuration on the subsequent nodes.

Example

Say you have a cluster that includes a primary Tableau Server node and three worker nodes, with gateway processes running on the primary, Worker 2 and Worker 3. In this situation, you configure the primary Tableau Server for SSL, and then copy the same SSL certificate and key files to Worker 2 and Worker 3, to the same location as on the primary.

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.

**Note:** Tableau Server only supports 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 install directory at the following location. Use this log to troubleshoot validation and encryption issues:

```
C:\ProgramData\Tableau\Tableau Server-\data\tabsvc\logs\httpd\error.log
```

Configure Internal SSL

This article applies to Tableau Server on Windows. For Tableau Server on Linux, see Configure SSL for Internal Postgres Communication.

You can configure Tableau Server to use Secure Sockets Layer (SSL) for encrypted communications on all traffic between the Postgres repository and other server components. By
default, SSL is disabled for communications between server components and the repository.

1. Open the Tableau Server Configuration Utility by selecting Start > All Programs > Tableau Server 10.5 > Configure Tableau Server.

2. In the Tableau Server Configuration dialog box, click the SSL tab.

3. Select one of the following options:

   - **Required for all connections**
     
     When this option is selected, Tableau Server uses SSL for communications between the repository database and other server components. In addition, direct connections to Tableau Server (connections using the "tableau" or "readonly" users) must use SSL.

   - **Optional for direct user connections**
     
     This option configures Tableau Server to use SSL between the repository and other server components and supports but does not require SSL for direct connections by "tableau" or "readonly" users.

   - **Off for all connections** (the default)
     
     This option disables SSL for internal communications and direct connections.

4. Click OK.

For more information on 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, and the REST API.

Before you can require SSL for direct connections, you need to enable Postgres (repository) SSL. For information, see Configure Internal SSL. If you know it’s enabled, complete the following steps:

1. Run the following command:

   ```
tabadmin regenerate_internal_tokens --certs
   ```

   This generates the server’s SSL certificate files.

2. Locate the certificate file.

   You can find the location in the workgroup.yml file. This file is located on the primary Tableau Server node, in the \ProgramData\Tableau\Tableau Server-\data\tabsvc\config folder.

   The location of the SSL certificate and key files are listed in the file. For example:

   ```
pgsql.ssl.cert.file: C:/ProgramData/Tableau/Tableau Server-/data/tabsvc/config/pgsql/server.crt
pgsql.ssl.key.file: C:/ProgramData/Tableau/Tableau Server-/data/tabsvc/config/pgsql/server.key
   ```

3. On the computer that will make the direct connections, do the following:

   a. Copy **server.crt** to the computer.

   ```
Note: Do not copy **server.key** to the client computer. This file should reside only on the server.
   ```

   b. Import the certificate into the computer’s certificate store.
For information, use the documentation from the operating system manufacturer.

See also

`tabadmin regenerate_internal_tokens`

`tabadmin`

**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). If you have multiple sites on Tableau Server and want to use multiple IdP applications, see Configure Site-Specific SAML.

Looking for Tableau Server on Linux? See [Configure Server-Wide SAML for Tableau Server on Linux](#).

Before you configure Tableau Server for SAML, make sure your environment meets the SAML Requirements.

To configure Tableau Server on Windows for server-wide SAML:

1. Place the SAML certificate files in a folder named SAML, parallel to the Tableau Server 10.5 folder. For example:

   ```
   C:\Program Files\Tableau\Tableau Server\SAML
   ```

   This is the recommended location because the user account that runs Tableau Server has the necessary permissions to access this folder.

2. In the Tableau Server Configuration Utility, open the **SAML** tab.
Post-installation, you can find this utility in the Tableau Server 10.5 > Configure Tableau Server folder.

3. Select SAML authentication for the server and provide the location for each of the following:

**Tableau Server return URL**—The URL that Tableau Server users will be accessing, such as http://tableau_server. Using http://localhost is not recommended. Using 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 like, but it does not have to be your Tableau Server URL.

**SAML certificate file**—A PEM-encoded x509 certificate with the file extension .crt. This file is used by Tableau Server, not the IdP.

**SAML certificate key file**—An RSA or DSA private key file that is not password protected, and that has the file extension .key. This file is used by Tableau Server, not the IdP.

4. Click Export Metadata File, and save Tableau Server settings to an XML file with a name of your choice. Later, you will provide this information to your IdP.

**Note:** If you are configuring SAML for AD FS, instead of going to the next step, continue with the steps described in Step 3: Configure AD FS to accept sign-in requests from Tableau Server, in a separate article.

5. 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 file you exported from
Tableau Server earlier in these steps.

- Confirm that your IdP uses **username** as the attribute element to verify.

6. Still within your IdP, export your IdP’s metadata XML file.

   It’s a good idea to verify that the metadata XML you get from the IdP includes a **SingleSignOnService** element in which the binding is set to **HTTP-POST**, as in the following example:


7. Copy your IdP’s metadata XML file to same folder on the Tableau Server computer that contains the SAML certificates:

   C:\Program Files\Tableau\Tableau Server\SAML

8. On the SAML tab in the Tableau Server Configuration dialog box, in the **SAML IdP metadata file** text box, enter the location of the file, and click **OK**.
Configure a Server Cluster for SAML

When you configure a Tableau Server cluster to use SAML, you place the same SAML certificate, SAML key, and SAML IdP metadata files on every computer that's running a Tableau application server process (also known as vizportal.exe). To configure a Tableau Server cluster to use SAML:

1. Configure the primary Tableau Server as described in the procedure above.

2. Place the same SAML certificate, SAML key, and SAML IdP metadata files that you used for the primary on each Tableau Worker that is running an application server process. Use the same folder location on the workers that you used on the primary. You do not need to do any additional configuration on the workers.

For example, consider a cluster that includes a primary Tableau Server and two workers. Application server processes run on the primary and on Worker 2 and Worker 3. In this situation, you configure the primary Tableau Server for SAML, and then copy
the same SAML certificate, SAML key, and SAML IdP metadata files to the Worker 2 and Worker 3 computers. On the worker computers, put the SAML files in the C:\Program Files\Tableau\Tableau Server\SAML folder, just as they are on the primary computer.

Test Your Configuration

1. In your web browser, open a new page or tab, and enter the Tableau Server URL.

The browser redirects you to the IdP's sign-in form.

2. Enter your single sign-on user name and password.

The IdP verifies your credentials and redirects you back to your Tableau Server start page.
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 you meet the meet 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.

1. Open a command prompt as an administrator and change directories to the location of Tableau Server's bin directory. The default location is `C:\Program Files\Tableau\Tableau Server\10.5\bin`.

2. Type the following command to stop Tableau Server:

   `tabadmin stop`

3. Open the Tableau Server Configuration Utility (Start > All Programs > Tableau Server 10.5 > Configure Tableau Server), and then click the Kerberos tab.

4. Select **Enable Kerberos for single sign-on**.

5. Click **Export Kerberos Configuration Script**. The generated script configures your Active Directory domain to use Kerberos with Tableau Server. For more information, see Kerberos Configuration Script.
Note: Verify the host names in the setspn lines of the script. If you are using an external load balancer or a reverse proxy, the host names should match the name you used when you configured Tableau Server for the load balancer or proxy. If you have not configured Tableau Server for your proxy or external load balancer, do that and then re-export the Kerberos configuration script to ensure it has the correct host names. See Add a Load Balancer and Configuring Proxies for Tableau Server.

6. Have your Active Directory domain administrator run the configuration script to create Service Principal Names (SPNs) and the .keytab file. The domain administrator should do the following:

   • Review the script to verify it contains correct values.
   
   • Run the script at a command prompt on any computer in the domain by typing the script name (not by double-clicking the script in Windows Explorer).

   The script creates a file, kerberos.keytab, in a \keytabs folder in the location that the script was run.

7. Save a copy of the .keytab file created by the script to the Tableau Server computer. In Step 3, enter the path to the .keytab file, or click the browse button to navigate to the file. The keytab file will be copied to all the gateway nodes in your Tableau Server installation when you click OK in the Configuration utility.

Note: Do not rename the .keytab file. The script creates a file named kerberos.keytab and you need to save it with this name.

8. (optional) Click Test Configuration to confirm that your environment is configured correctly to use Kerberos with Tableau Server.
If you have not configured any data sources for Kerberos delegation, 0 is shown for the **Number of services configured for delegation**.

9. Click **OK** to save your Kerberos configuration.

10. Start Tableau Server.

**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:

![Web browser with URL](image)

You should be automatically authenticated to Tableau Server.

**Configure SAP HANA SSO**

You can configure Tableau Server to use SAML delegation to provide Single Sign-on (SSO) for SAP HANA. HANA SSO is not dependent on SAML authentication to Tableau Server. **Note:** You do not need to use SAML sign on with Tableau Server in order to use HANA SSO. You can sign in to Tableau Server using whatever method you choose.
With SSO for SAP HANA, Tableau Server functions as an Identity Provider (IdP) and this configuration allows you to provide a single sign-on experience for users making SAP HANA connections. As part of the configuration, you need to acquire a SAML certificate and key file for Tableau Server (these should be a public key certificate and private key). You need to also install the signed certificate in HANA. You can generate the certificate and key yourself, or get them from a Certificate Authority. For more information on generating a certificate/private key and configuring SAP HANA, see the Tableau Community.

**Note:** The SAP HANA driver version 1.00.9 or later must be installed on Tableau Server in order to use SSO for SAP HANA. The driver cannot encrypt the SAML assertion, so you may want to enable encryption for the SAML connections. For more information, see the Tableau Community.

**Configure SSO for SAP HANA**

To configure Tableau Server to use SSO for SAP HANA:

1. Place certificate files in a folder named SAML, parallel to the Tableau Server 10.5 folder. For example:

   `C:\Program Files\Tableau\Tableau Server\SAML`

   You should use this location because the user account that runs Tableau Server has the necessary permissions for accessing this folder.

2. After you install Tableau Server, run the Configuration utility (Start > All Programs > Tableau Server 10.5 > Configure Tableau Server), and then click the SAP HANA tab.

3. Select **Use SAML to enable single sign-on for SAP HANA** and provide the location for each of the following:
SAML certificate file—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.

SAML private key file—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.

4. Select the format of the user name.

5. Select the case for the user name. This determines the case of the name when it is forwarded to the SAP HANA identity provider (IdP).

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 (IdP) for OpenID Connect
- Configure Tableau Server for OpenID Connect (you are here)
- Signing In to Tableau Server Using OpenID Connect
- Changing IdPs in Tableau Server for 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 (IdP) for OpenID Connect.

**Important notes**

Before you configure Tableau Server for OpenID Connect, make sure you read these notes.

- You can use OpenID Connect with Tableau Server only if the server is configured to use local authentication. OpenID Connect is not available if the server is configured to use Active Directory authentication. For more information, see Configure General Server Options.

- If you are configuring Tableau Server for the Salesforce IdP, then you must set the `vizportal.openid.client_authentication` parameter. See `tabadmin` configuration options for more information.

- We recommend that you configure Tableau Server to use SSL for external communications. This helps to maintain secure communications between Tableau Server and the IdP during the exchange of authentication information. For details, see Configure External SSL.

  If your IdP uses a self-signed certificate (or the IdP certificate is not signed by a trusted CA), then you can add the IdP public certificate to the Java truststore. See
Configuring Tableau Server for OpenID Connect Using Uncommon or Self-Signed Certificates.

If you are configuring OpenID Connect during the initial configuration of Tableau Server (the first time the configuration utility runs), there is no option to set up SSL. In that case, we recommend that you finish the installation, then return to the configuration to set up SSL and then configure OpenID.

**Note** If you want to use external SSL for Tableau Server, it's generally more convenient to do that before you configure OpenID Connect. If you configure SSL after you've already configured OpenID, you need to return to the IdP and update the configuration that you made previously. For example, you need to change the protocol for the Tableau Server external URL from http:// to https://.

Configure the server

To configure Tableau Server for OpenID Connect, follow these steps.

1. Log in as an administrator to the computer where Tableau Server is running.

2. If the server is running, stop it (Windows Start > All Applications > Tableau Server > Stop Tableau Server).

   **Tip:** You can also stop the server by using the tabadmin stop command.

3. Run the Tableau Server Configuration tool (Windows Start > All Applications > Tableau Server > Configure Tableau Server).

4. Click the OpenID tab.

5. Select the Use OpenID Connect for single sign-on option.

6. Fill in the Provider client ID and Provider client secret boxes with the values you
recorded earlier.

7. In the Provider configuration URL box, enter the URL that the IdP uses for OpenID Connect discovery.

Alternatively, you can configure Tableau Server to reference a static discovery document to configure OpenID. The discovery file is a JSON document that defines the IdP configuration. Copy the file to the Tableau Server and then run the following sequence of tabadmin commands:

```
tabadmin stop
tabadmin set vizportal.openid.static_file <file-path>
tabadmin configure
tabadmin start
```

Where `<file-path>` is the full path of the discovery file, for example, `c:\files\openid-configuration`.

8. In the Tableau Server external URL box, enter the URL of your server. This is typically the public name of your server, such as `http://example.tableau.com`.

When you initially configure OpenID, the Provider configuration URL box contains a default value that’s 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.

**Note:** Make sure that you update the external URL if the default value is not the URL for how your server can be reached from an external source.
9. Copy the URL in the box labeled **Configure the OpenID provider using the following redirect URL for Tableau Server**. You’ll use this value in the next procedure to finish configuring the IdP.

10. Start the server (Windows Start > All Applications > Tableau Server > Start Tableau Server).

    **Tip:** You can also start the server by using the tabadmin start command.

Add the redirect URL to the IdP configuration

After you configure Tableau Server, you finish the IdP configuration using the server’s redirect URL.

1. Return to the IdP portal where you set up the project or application.

2. Edit the project configuration and find the redirect URL.

3. Enter the redirect URL that you copied in the previous procedure.
Add an Administrator Account

The final step in activating Tableau Server is to add an administrator account. The administrator will have all access to the server including the ability to manage users, groups, and projects. Adding an administrator account differs depending on whether you are using Active Directory or local authentication.

Active Directory

If you are using Active Directory, type the **Username** and **Password** for an existing Active Directory user who will be the administrator. Then click **Add user**.

![Tableau Server Setup Tasks](image)

### Note:

If the administrator account is in the same domain as the server simply type the username without the domain. Otherwise you should include the fully qualified domain name. For example, test.lan\username.

Local Authentication
If you are using Local Authentication, create an administrative account by typing a **Username**, **Display Name**, and a **Password** (twice) of your choosing. Then click **Add user**.

### Tableau Server Setup Tasks

**Add Administrator Account**

<table>
<thead>
<tr>
<th>Username:</th>
<th>Administrator</th>
</tr>
</thead>
<tbody>
<tr>
<td>Display Name:</td>
<td>Administrator</td>
</tr>
<tr>
<td>Password:</td>
<td>****</td>
</tr>
<tr>
<td>Confirm password:</td>
<td>****</td>
</tr>
</tbody>
</table>

Click **Add user**

---

**Reconfigure the Server**

When you install Tableau Server for the first time, you do initial configuration of the server as part of the installation. You can run the Tableau Server Configuration utility after installing Tableau Server to make additional configuration changes. Some configuration options are only available when you run the configuration utility after installation. You can also use the `tabadmin` command line tool to make configuration changes. **Configuration setting changes are written to the `tabsvc.yml` file located in the `<install drive>:\ProgramData\Tableau\Tableau Server\config` directory.**

**Note:** You cannot switch between Active Directory and Local Authentication. These options can only be configured during the initial installation of Tableau Server.
To change a Tableau Server configuration setting:

1. Stop the server by selecting **All Programs > Tableau Server 10.5 > Stop Tableau Server** on the Windows Start menu.

2. Select **Configure Tableau Server** on the Windows Start menu.

3. If you are using an Active Directory account for the server's Run As User account, enter its password on the **General** tab.

4. Make your configuration change.

5. Click **OK**.

6. Start the server by selecting **All Programs > Tableau Server 10.5 > Start Tableau Server** on the Windows Start menu.

**Reconfigure Processes**

To change how processes are configured for a single server installation, follow the steps below. If you are changing how processes are configured for a worker, refer to Install and Configure Worker Nodes.

1. You will need to stop Tableau Server to make this configuration change. From the Start menu, click **All Programs > Tableau Server 10.5 > Stop Tableau Server**.

2. Open the Tableau Server Configuration dialog box from the Start menu by navigating to **All Programs > Tableau Server 10.5 > Configure Tableau Server**.

3. Enter your **Password**, if necessary, on the **General** tab then click the **Servers** tab:
4. **Highlight This Computer and click Edit:**

5. The **Edit Tableau Server** dialog box is where you change the number of processes:
You can run up to eight instances of the VizQL, application server, data server, or background processes—although this limit can be changed if necessary. See Server Process Limits for more information. You need to have at least one instance of backgrounder installed. Also, for Tableau Server to function, there must always be one active instance of the file store and the repository. For steps on how to move them to another machine, see Move the File Store Process. For steps on how to configure additional instances of them, refer to High Availability.

After you make your changes, click **OK**.

6. If you want to designate a specific computer as the preferred active repository, select the computer from the **Select host** list. If you add workers, you need to save the
configuration and restart the Configuration utility for the workers to display in the list.
For more information about the repository, see Tableau Server Repository.

7. Click **OK** to close the Configuration utility.

8. Start Tableau Server again. From the Start menu, click **All Programs > Tableau Server 10.5 > Start Tableau Server**.

Tableau Server Processes

Tableau Server installs a number of processes that work together to deliver the features that make up Tableau Server.

Looking for Tableau Server on Linux? See **Process Reference**.

In this article

- Configuring processes
- Licensed processes
- List of processes

Configuring processes

Certain processes listed below cannot be configured: cluster controller and coordination service are installed on every node as part of the base install. They are required on every server node and do not count against a core-based license. Data engine is installed when you install file store, Vizportal, VizQL Server, data server, or backgrounder, and cannot be installed separately. Every node that has an instance of one of those processes will also have a single instance of the data engine process.

The topics Performance Tuning Examples and High Availability describe some of the approaches you can take when configuring processes. High-level status for each process is displayed on the server’s Status page and more detailed information related to some of the processes—such as the background process—is in the Administrative Views topic.
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 primary Tableau Server computer to verify they are licensed. If they cannot confirm there is a valid license, for example, if the primary 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.

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.

List of processes

For information on log files generated by these processes, see Server Log File Locations.

<table>
<thead>
<tr>
<th>Process</th>
<th>Purpose</th>
<th>Multi-Threa-</th>
<th>Performance Characteristics</th>
<th>Licensed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Application Server vizportal.exe</td>
<td>Handles the web application, REST API calls, supports browsing and searching</td>
<td>Yes</td>
<td>Only consumes noticeable resources during infrequent operations, like publishing a workbook with an extract, or generating a static image for a view. Its load can be created by browser-based interaction and by tabcmd.</td>
<td>Yes</td>
</tr>
<tr>
<td>Process</td>
<td>Purpose</td>
<td>Multi-Threaded</td>
<td>Performance Characteristics</td>
<td>Licensed</td>
</tr>
<tr>
<td>--------------------</td>
<td>------------------------------</td>
<td>----------------</td>
<td>-----------------------------------------------------------------------------------------------</td>
<td>----------</td>
</tr>
<tr>
<td>Backgrounder</td>
<td>Executes server tasks, including extract refreshes, subscriptions, 'Run Now' tasks, and tasks initiated from tabcmd</td>
<td>No</td>
<td>A single-threaded process where multiple processes can be run on any or all machines in the cluster to expand capacity. The backgrounder normally doesn’t consume much process memory, but it can consume CPU, I/O, or network resources based on the nature of the workload presented to it. For example, performing large extract refreshes can use network bandwidth to retrieve data. CPU resources can be consumed by data retrieval or complex tabcmd tasks.</td>
<td>Yes</td>
</tr>
<tr>
<td>Cache Server</td>
<td>Query cache</td>
<td>No</td>
<td>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 API server and application server to a lesser extent) make cache requests to 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>No</td>
</tr>
<tr>
<td>Cluster Controller</td>
<td>Responsible for monitoring</td>
<td>n/a</td>
<td>Included in the base install on every node.</td>
<td>No</td>
</tr>
<tr>
<td>Process</td>
<td>Purpose</td>
<td>Multi-Threaded</td>
<td>Performance Characteristics</td>
<td>Licensed</td>
</tr>
<tr>
<td>--------------------------</td>
<td>-------------------------------------------------------------------------</td>
<td>----------------</td>
<td>---------------------------------------------------------------------------------------------</td>
<td>----------</td>
</tr>
<tr>
<td>Coordination Service</td>
<td>In distributed installations, responsible for ensuring there is a quorum for making decisions during failover</td>
<td>n/a</td>
<td>Always installed on the primary node. For server installations with three to five nodes, also installed on the first two worker nodes. For server installations of more than five nodes, also installed on the first four worker nodes.</td>
<td>No</td>
</tr>
<tr>
<td>hyperd.exe</td>
<td>Creates extracts and processes queries.</td>
<td>Yes</td>
<td>The data engine’s workload is generated by requests from the VizQL server, application server, API server, data server, and backgrounder server processes. The data engine services requests from most of the other</td>
<td>Yes</td>
</tr>
<tr>
<td>Process</td>
<td>Purpose</td>
<td>Multi-Threaded</td>
<td>Performance Characteristics</td>
<td>Licensed</td>
</tr>
<tr>
<td>------------------</td>
<td>--------------------------------------------</td>
<td>----------------</td>
<td>-----------------------------------------------------------------------------------------------</td>
<td>----------</td>
</tr>
<tr>
<td>Data Server</td>
<td>Manages connections to Tableau Server data sources</td>
<td>Yes</td>
<td>Because it’s a proxy, it’s normally only bound by network, but it can be bound by CPU with enough simultaneous user sessions. Its load is generated by browser- and Tableau Desktop-based interaction and extract refresh jobs for Tableau Server data sources.</td>
<td>Yes</td>
</tr>
<tr>
<td>File Store</td>
<td>Automatically replicates extracts</td>
<td>n/a</td>
<td>A data engine is automatically installed when file store is installed, unless the node already has an instance of data engine.</td>
<td>No</td>
</tr>
</tbody>
</table>

The data engine is automatically installed when you install file store, VizQL Server, Application Server (VizPortal), Data Server, or Back grounder. You cannot install Data Engine separately.
<table>
<thead>
<tr>
<th>Process</th>
<th>Purpose</th>
<th>Multi-Threaded</th>
<th>Performance Characteristics</th>
<th>Licensed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Repository postgres.exe</td>
<td>Tableau Server database, stores workbook and user metadata</td>
<td>n/a</td>
<td>Normally consumes few resources. It can become a bottleneck in rare cases for very large deployments (thousands of users) while performing operations such as viewing all workbooks by user or changing permissions. For more information, see Tableau Server Repository.</td>
<td>No</td>
</tr>
<tr>
<td>Search &amp; Browse search-server.exe</td>
<td>Handles fast search, filter, retrieval, and display of content metadata on the server</td>
<td>Yes</td>
<td>The process is memory bound first, and I/O bound second. The amount of memory used scales with the amount of content (number of sites/-projects/workbooks/datasources/views/users) on the server.</td>
<td>No</td>
</tr>
<tr>
<td>VizQL Server vizqlserver.exe</td>
<td>Loads and renders views, computes and executes queries</td>
<td>Yes</td>
<td>Consumes noticeable resources during view loading and interactive use from a web browser. Can be CPU bound, I/O bound, or network bound. Process load can only be created by browser-based interaction. Can run out of process memory.</td>
<td>Yes</td>
</tr>
</tbody>
</table>
Tableau Server Coordination Service

Tableau Server uses the Coordination Service to coordinate activities on the server, including for high availability installations. The Coordination Service is built on Apache ZooKeeper, an open-source project.

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.

Configuration for the Coordination Service

The Coordination Service is installed automatically as a part of Tableau Server. The number of nodes with the Coordination Service installed depends on the total number of nodes in the Tableau Server installation. On a one- or two-node system, one instance of Coordination Service is installed. On a three- or four-node installation, three instances of Coordination Service are installed. For Tableau Server installations that have five or more nodes, a total of five Coordination Service instances are installed. (The Coordination Service is installed on the first <n> nodes in the cluster, so on a one- or two-node cluster it is installed on the first node, on a three- or four-node cluster it is installed on the first three nodes, and on a cluster of five or more nodes it is installed on the first five nodes.)

You do not have to explicitly configure the coordination service, and there are no settings you can make for the service. As a consequence, when you add a node to your cluster, you
do not see the configuration service listed as process—for example, you do not see the coordination service listed in the **Add Tableau Server** dialog box:

![Edit Tableau Server](image)

The **Base Install** option includes the Coordination Service and Cluster Controller. As you can see, this option is disabled, because you cannot choose when to install those services.

**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 you have in your installation impacts how many instances of the Coordination Service are running.
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 following tabadmin command:

tabadmin status --verbose

The output from the command shows you whether the service is running:

10.32.139.21:
   Status: RUNNING
   'Tableau Server Data Engine 0' (2456) is running.
   'Tableau Server Vizqlserver 0' (3336) is running.
   'Tableau Server Backgrounder 0' (11976) is running.
   'Tableau Server CacheServer 0' (2508) is running.
   'Tableau Server Dataserver 0' (3572) is running.
   'Tableau Server Application Server 0' (804) is running.
   'Tableau Server API Server 0' (3584) is running.
   'Tableau Server Coordination Service 0' (2624) is running.
   'Tableau Server Search and Browse 0' (2744) is running.
   'Tableau Server Gateway' (2824) is running.
   'Tableau Server Cluster Controller' (2840) is running.
'Tableau Server Repository' (2032) is running (Active Repository).
'Tableau Server File Store' (2964) is running.

Performing Cleanup for the Coordination Service

The Coordination Service maintains state information about the server, such as transaction logs of activities on the server. This information is written to disk, and when the server is restarted, the information on disk is used to restart the Coordination Service and to determine state information such as whether multiple repositories have been synchronized.

If the data maintained by the service is corrupted (for example, due to hardware problems) or if there is some other problem with the Coordination Service that affects server startup, you can perform a cleanup operation on the service's information. To do so, run the following `tabadmin` command:

```
tabadmin cleanup --reset-coordination
```

This command will perform a normal cleanup as well as removing Coordination Service files.

**Note:** This command can only be run when the server is stopped.

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>File name</td>
<td>filestore.exe</td>
</tr>
<tr>
<td>Status</td>
<td>Status of the File Store process is visible on the Status Page. For more inform-</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 decommission command. 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.

Tuning the Cluster Controller Timeout Threshold

Tableau Server uses a coordination service (built on Apache ZooKeeper) to maintain configuration information and manage synchronization across the components in Tableau Server. This topic discusses the behavior of the coordination service, how that can affect repository and failover behaviors under some circumstances. This topic also describes how to tune the coordination service, if necessary, to the hardware performance of the computer that Tableau Server is running on.

Coordination Service and disk I/O

The Coordination Service process writes Tableau Server configuration information to disk. Therefore, disk I/O performance is critical for the service. If disk I/O falls below a certain threshold (by default, 5 megabytes per second), Tableau Server restarts the Coordination Service process.

A number of factors may cause the Disk I/O performance to drop below the 5 MB/sec threshold:

- Tableau Server is running on hardware at or below the low end of the recommended hardware configuration
• Tableau Server is installed on a computer where other disk I/O-intensive software is running
• Tableau Server is processing disk I/O-intensive operations, such as publishing large extracts

In these cases, Tableau Server might restart the Coordination Service process.

Coordination Service and the Cluster Controller timeout

The Cluster Controller process monitors the health of services across the cluster, including the coordination service. The Cluster Controller process has a timeout value set to five minutes (versions prior to 9.0.5 default to one minute).

After this interval, the Cluster Controller process triggers a failover from the active repository to the passive repository. On a single-node Tableau Server installation, the Cluster Controller process triggers a restart of the Repository process.

This behavior is by design. Tableau Server does not allow a repository to get too much out of date and will switch to a more up-to-date repository to allow the out-of-date one to get caught up.

Tuning the Cluster Controller timeout threshold

A long (five minute) timeout threshold makes Tableau Server less sensitive to temporary coordination service outages. If the service stops due to a brief spike in disk I/O, then restarts before the end of the timeout period, Tableau Server remains available and does not need to failover.

However, a longer timeout period can have implications for systems that failover due to issues other than temporary coordination service unavailability. The increased timeout period creates a longer window during which information written to the active repository could be lost if a failover takes place.

If you are running robust hardware and you have not been seeing disk I/O bottlenecks or performance issues, then you should consider setting a shorter timeout threshold for the Cluster Controller. Setting a shorter interval limits the potential data loss window by reducing the
time during which the active repository is updated without a corresponding update to the passive repository.

If you are seeing disk I/O bottlenecks, do not decrease the timeout threshold, or do so in small increments. As discussed above, one circumstance that can cause a failover is when the Coordination Service process is not available because it is restarting due to low disk I/O performance. In that case, the repository failover occurs because of a lack of reporting from the Coordination Service, not because the repository is truly out of date. On underpowered hardware, especially hardware with limited disk I/O, the coordination service might temporarily drop its connection. As a result, tuning the timeout threshold too low will result in frequent failovers.

**To reset the timeout**

Run the following sequence of tabadmin commands

```
tabadmin set clustercontroller.zk_session_timeout_ms 60000
```

```
tabadmin config
tabadmin restart
```

This sets the timeout to 60000 milliseconds (60 seconds).

**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.
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, you can change the port assignment for the gateway process. You can do this in the Tableau Server Configuration tool:
Alternatively, you can run the following `tabadmin` command, where `nn` is the new port number:

```
tabadmin gateway.public.port nn
```

Log files for the gateway process

The gateway process creates two sets of log files in the `\logs\httpd` folder of the log file archive:

- **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 Archive Log Files.

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. For more information about using SSL, see Configure External SSL.

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:

```
tagadmin workerN.gateway.port nn
```

Additional information

Configuring Proxies for Tableau Server

Add a Load Balancer

Configure for Failover and Multiple Gateways

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.
<table>
<thead>
<tr>
<th>Process</th>
<th>Repository</th>
</tr>
</thead>
<tbody>
<tr>
<td>File name</td>
<td>postgres.exe</td>
</tr>
<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>
<tr>
<td>Logging</td>
<td>Logs generated by the repository are located in \logs\repository. For more information, see Server Log File Locations</td>
</tr>
</tbody>
</table>

Preferred active repository

When you configure Tableau Server after the initial installation, you have the option to specify a **Preferred Active Repository**. This is an optional step, and if you do not specify a preferred active repository, Tableau Server will select the active repository on startup.

Configure a preferred active repository 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
failoverrepository 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 failoverrepository.

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. After Setup, stop the server.

2. In the Tableau Server bin directory, type the following command, where number is the maximum number of process instances you want to allow:

   tabadmin set service.max_procs <number>

   For example:

   tabadmin set service.max_procs 10

3. Still in the bin directory, type:

   tabadmin config

4. Start the server so the changes can take effect.
Changing Authentication

Infrastructure or business changes may require you to change the authentication type on Tableau Server. In this context, authentication refers to the core authentication type that you’ve configured on Tableau Server: local authentication or Active Directory. See Authentication to learn more.

You can change from local authentication to Active Directory or you can change from Active Directory to local authentication. In either case, to change the authentication 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.

Warning

Changing the authentication 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.

Most importantly, determine how you will transition content and permissions to the new authentication model after you reinstall Tableau Server.

Methods for restoring content and permissions

The following list describes three 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 authentication 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 authentication change is complete. In most cases where organizations need to change the authentication type, a different user name syntax is often a requirement. 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 authentication method 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 methods, 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.

- **Method 3: Restore and manually assign content**—In this method, you begin by documenting the content ownership model. Then you take a backup of the server, uninstall the existing Tableau instance, delete the install directory and then install a new instance with the new authentication type. After installation, you manually create new users that correspond to users on the original system. You then restore the backup to upload content. Lastly, you reassign content to users as appropriate. This method requires careful planning and implementation to avoid orphaned users and content.

  This method is best if you are archiving more content than you are reassigning. In this context, "archiving" content means that you’re reassigning old content to a single
administrative user rather than to individual users. On the other hand, if all (or most) of your users need to maintain control over the content that they have on the existing system, then you must manually reassign all of the content to match the original ownership model. In addition to the cost of reassigning content after the change, you must also document content ownership before the change.

**User names and the Tableau Identity store**

Unless you select Method 2 above, 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, servers configured with Active Directory authentication store user names in the format, domain\username. Some organizations use a UPN (name@-domain.lan). On the other hand, organizations that configure Tableau Server with local authentication usually create standard, truncated user names, such as jsmith.

In all cases, these user names are literal strings that must be unique in Tableau identity store for the purpose of identity management. If you are changing from one authentication 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 authentication type:

- The new user names must match the original user names.
- The original user names must be updated to match a new format.

In most cases, your new authentication plan will impose a user name syntax that is different than your original user names. In that case, we recommend using Method 1 - site import to change authentication type. The site import method 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 authentication 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 authentication, as long as users continued to use that format to sign in to Tableau Server. For these cases, Method 3 might be the most efficient way to change the authentication type.

If you are changing from local authentication to Active Directory authentication, 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, uninstall, 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 authentication change, we recommend that you back up the server.

1. Back up, uninstall, 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.

Method 3: Restore and manually assign content

Before you begin this method, we recommend that you document the existing deployment. Specifically, gather the following information from the existing server:

- User information: user names, display names
- Sites: site names and users in each site
- Groups: group names and permissions
- Content: workbooks, views, data sources

After you have gathered this information, follow these steps:
1. Follow the procedures detailed below (Back up, uninstall, and then reinstall).
   However, when you are prompted to create a new administrative account, close the browser without creating the account. You will create a new administrative account later in this procedure.

2. Open Windows Command Prompt as an administrator and navigate to the Tableau Server bin directory by typing the following command:

   ```
   cd C:\Program Files\Tableau\Tableau Server\<version>\bin
   ```

3. Type the following command to restore the backup file without inheriting the previous configurations:

   ```
   tabadmin restore [path\file name] --no-config
   ```

   After restoring your backup file, a browser window will display and you will be prompted to create a new administrator account again. Ignore this, and close the browser window. If you are prompted to enter the Run As User password, type the password and continue with the next step. You will not see a password confirmation.

4. In the Command Prompt window, from the Tableau Server bin directory, type the following command to reset Tableau Server back to a state that requires an administrator account to be created:

   ```
   tabadmin reset
   ```

5. Open a browser window, and enter http://localhost in the address bar to create the administrator account for Tableau Server.

   If you are using Active Directory authentication, the account for the Tableau Server administrator account must be an existing Active Directory user.

6. Sign in to Tableau Server. Users from your previous installation have been migrated. These users are not valid and will need to be deleted.

7. Create new users and assign content to those users as appropriate.
Back up, uninstall, and then reinstall

All methods include the following steps:

1. Back up Tableau Server
2. Uninstall Tableau Server.
3. Reinstall Tableau Server with the new authentication type.

Step 1: Back up Tableau Server

Follow the procedure, Creating a regular backup. Run the `backup` command with the `-d` and the `-v` options. The `-d` option adds the datestamp, and the `-v` option verifies the state of the database for backup and restore.

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: Uninstall Tableau Server

1. From the Start menu, go to Control Panel > Programs and Features. Select Tableau Server, and then click Uninstall.
2. When Tableau Server has been uninstalled, open Windows Explorer and navigate to \Program Files\Tableau.
3. Delete the Tableau Server folder.
4. While still in Windows Explorer, navigate to \ProgramData\Tableau (this is a hidden directory), and then delete the Tableau Server folder.

   **Note:** If you have installed Tableau Server on a drive other than the system drive, the folder paths will be different than those shown here. See Installation directory for more information.

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 for more information. During installation, you will select the new authentication method. See Configure General Server
Options. If you are running Method 3, then ignore the prompts (close the browser window) to create a new administrative account on reinstall.

Distributed Environments

Quick Start: Distributed Server

Increase the scalability of your Tableau Server environment by distributing the server components across several machines. Install Tableau Server on your primary node, then use the Worker Installer to install the software onto one or more “worker” nodes. With the worker nodes installed, configure the primary node to use those workers.

1 Install Tableau Server on Worker Nodes

Download the Tableau Server Worker installer from the download site and install it on all of the computers (other than the primary server) that you want to include in the distributed installation of Tableau Server. Be sure to download the same version of the worker software that you installed on the primary node.

Install Tableau Server on the primary node before you install on the other nodes in the distributed installation.

2 Open Configuration Utility

On your primary node, open the configuration utility by selecting Tableau Server 10.5 > Configure Tableau Server.
3 Add Worker Nodes

In the Configuration Utility, select the **Servers** tab and click **Add**. In the Add Tableau Server dialog box, type the IP address or the name of the worker node in the **Computer** box and
specify the number of processes to allocate to the node. Repeat this for each machine you want to include in the distributed installation.

You can click **Discover** to automatically add any worker computers you installed in step 1 above.

### 4 Configure & Update Primary Node

After you set up the worker nodes, make all configuration changes and updates on the primary node. Use the command line tools and the Tableau Server Configuration utility on the primary node. Updates will be pushed to the workers automatically.

### 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 primary Tableau Server and two for Tableau worker nodes.
Note: If you install Tableau Server on a two-node cluster, a message displays to let you know that you are limited to one instance of the repository, and that high availability and failover are not available in a two-node configuration. You can add a third node but are not required to do so. In a two-node cluster, if one of the two nodes goes down, Tableau Server may not function correctly.

- **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—four computers and a load balancer:** To configure for high availability, you need the resources described above plus an additional computer to be the backup primary for your primary Tableau Server.

- **Primary computers:** If you configure for high availability, the primary Tableau Server and the backup primary may be running few or no Tableau Server processes. Therefore, the computers that run the primary and backup primary do not need as many cores as the ones running your worker servers. You will, however, need adequate disk space for backups because the primary 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**

As of version 10.0, Tableau Server is only available in a 64-bit version.

Earlier versions were offered in both 32-bit and 64-bit versions. If you are running a version of Tableau Server that was available in both 32-bit and 64-bit, be aware that each computer must run the bit version—either all 64-bit or all 32-bit. For example, if the primary Tableau Server is running the 64-bit version of Tableau Server, the workers in the cluster must run the 64-bit version of Tableau Server Worker.
Installation location

- The installation location for Tableau Server must be the same on all nodes in a cluster. This is true whether you install to the default location or to a non-default location.

Networking and Ports

- **Ports:** As with any distributed system, the computers or VMs you use need to be able to communicate with one another. See Tableau Server Ports for a list of ports that must be available on the gateways and workers.

- **Same domain:** If Tableau Server is installed in a Windows Active Directory environment, then all computers in a cluster must be members of the same domain.

- **Same subnet:** Do not install a distributed system across multiple subnets. Each node in the cluster must be installed on the same subnet.

- **Service account:** The server’s Run As User account, which is specified on the primary Tableau Server, must be the same on each computer in the cluster. If you are not running in an Active Directory environment, then we recommend updating the Run As User with a Windows workgroup user. You must specify the same user account and password on each node in the cluster. While you can leave the default NetworkServices account on each node in the cluster, we do not recommend this as a best security practice.

- **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 (learn more).

- **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.

- **Task Scheduler**: The Windows Task Scheduler service must be running on each node or the installation will fail with an error like "Network Address Invalid".

**Best Practices**

Here are some things to keep in mind before you start to install and configure:

- **IP addresses or computer names**: Note the IPv4 addresses or computer names of each computer or VM you'll be working with. You will need to provide them during Tableau Worker Setup and configuration. As mentioned above, each computer in the cluster must use a static IP address, even if you use the computer’s name to identify it during configuration.

- **CNAME record**: If you’re configuring for high availability and you are not using a load balancer, make sure your primary Tableau Server and backup primary have the same CNAME record so that your Tableau Server users have a smooth experience if one primary fails and you configure the other to take over. If you are using a load balancer, it's the load balancer’s name that users will be using as the Tableau Server URL, regardless of the gateway that's actually handling the request.

- **User account credentials**: For each computer, you need credentials for a user account with local admin permissions. If you’re configuring for high availability, the Run As account you use for your primary Tableau Server must be the same as the one you use for your backup primary Tableau Server.

- **Backup**: It’s a best practice to create a backup prior to making significant system changes. See Back Up Tableau Server Data for steps.

- **Distributed deployment across data centers**: We do not recommend installing Tableau Server nodes across distributed data centers. The following examples describe some issues that are common when nodes are distributed between multiple data centers:
- Disruption in network connectivity between nodes can cause many tasks to fail or for Tableau Server components to become unlicensed.
- Proxies and firewalls between data centers may impede the ability of the Tableau Server nodes to communicate with each other.
- Routing traffic between geographically dispersed data centers, can cause latency or bottleneck data transmission, resulting in poor performance and connection timeouts.

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 a Cluster for other details.

Distributed Installation Recommendations

When you add computers (workers) 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 workers, see Distributed Environments.
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 during setup with the Backgrounder process. 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.

  **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 Worker Nodes

After you complete the initial configuration, you can set up Tableau Server to run on multiple computers. This is called a distributed installation, or cluster.

Running a distributed installation uses additional ports on the primary Tableau Server and requires that certain ports be available for binding during Setup on the Tableau Worker Servers. See Tableau Server Ports for more information. There are also additional requirements to be aware of when you run a distributed installation. See Distributed Requirements for details.

**Note:** If you install Tableau Server on a two-node cluster (the primary and one worker) with a file store on each node, a warning displays to let you know that you will not have failover support with this configuration and asking if you want to add a third node. You are not required to add a third server to the cluster, but with a two-node cluster there is no failover support, and if one of the two nodes goes down, Tableau Server will shut down. You cannot install multiple instances of the repository in a two-node cluster. For more information, see Two-node installations are limited to a single instance of the repository.

To install Tableau Server worker nodes:

1. Make sure you’ve installed Tableau Server on the primary computer.
2. Stop Tableau Server on the primary node (see Tableau Server Monitor to learn how).
3. Download the Tableau Server Worker software from the Tableau Customer Account Center.
4. Run Tableau Server Worker Setup on each computer you want to add to the Tableau Server cluster.

5. During installation you will be asked to provide the IPv4 addresses or computer name of the primary server. Using a computer name is recommended. If the primary has multiple network interface cards (NICs) enabled and you choose to enter IPv4 addresses, enter all of the primary’s IPv4 addresses, separating each with a comma. The IP address(es) for the computer running the primary must be static, this applies even if you use a computer name to identify the primary (learn more).

If you have a worker running Windows 7 with Windows Firewall enabled, you may get an error ("Worker is not responding") when you attempt to add a worker node. To avoid this issue, disable Windows Firewall on the worker computer before adding the worker to the Tableau Server cluster.

**Note:** If you configured SAML on the Tableau Server primary node, you need to copy the SAML certificate, SAML key, and SAML IdP metadata files to each node that’s running a Tableau application server process (vizportal.exe). For more information, see Configure a Server Cluster for SAML.

To configure Tableau Server for worker nodes:

1. Once the Worker software is installed on worker computers, and with the primary Tableau Server still stopped, return to the primary server and open the configuration utility by selecting **Tableau Server 10.5 > Configure Tableau Server** on the Start menu.

2. In the Configuration Utility, enter your password on the **General** tab then select the **Servers** tab and click **Add**.
3. In the next dialog box, type the IPv4 address or computer name for one of the worker computers and specify the number of VizQL, Application Server, Backgrounder, Cache Server, and Data Server processes to allocate to the computer.

With the 64-bit version of Tableau Worker Server, you can run up to two instances of each process. In rare cases and if the server’s hardware allows, that limit can be changed. See Server Process Limits and Tableau Server Performance Overview for more information.

Select File Store, Repository, Gateway, and Search & Browse processes to install one instance of each of these processes on the computer. One instance of Data engine (Hyper) is installed when you install file store, Vizportal, VizQL Server, 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 data engine (Hyper) process.

**Note:** You can only add a second repository if you have three or more nodes. If you attempt to add a repository to a two-node cluster, a message displays to let
you know that you are limited to one instance of the repository, and that high availability and failover are not available in a two-node configuration.

By default, the file store, repository, and gateway are hosted on the primary server. Running these processes on an additional server, or moving them off of the primary server, is part of configuring for high availability. See High Availability for more information.

4. Click **OK**. It may take several minutes for the updates to complete.

5. Repeat these steps for each computer you want to add to the distributed environment.
When you’re finished adding workers, click **OK** to save the changes and close the Configuration utility, then start Tableau Server on the primary node.

**Database Drivers**

The Tableau Server installer includes drivers for Oracle, Oracle Essbase, and the PostgreSQL databases. If you plan to publish workbooks and data sources that connect to other databases, you will need to install the corresponding drivers.

You need to install all database drivers on the initial node in a Tableau Server cluster. Additional nodes that run VizQL, application server, data server, or backgrounder processes also need these drivers. For example, if you dedicate one subsequent node to run as a VizQL server, and dedicate another computer for extract storage, you need to install drivers only on the computer running the VizQL server process.

The following table helps you determine whether you need to install drivers on a subsequent node in a cluster.

<table>
<thead>
<tr>
<th>Server process</th>
<th>Requires database driver?</th>
</tr>
</thead>
<tbody>
<tr>
<td>VizQL server</td>
<td>yes</td>
</tr>
<tr>
<td>Application server</td>
<td>yes</td>
</tr>
<tr>
<td>Data server</td>
<td>yes</td>
</tr>
<tr>
<td>Backgrounder</td>
<td>yes</td>
</tr>
<tr>
<td>API server</td>
<td>yes</td>
</tr>
<tr>
<td>Data engine (extract storage)</td>
<td>no</td>
</tr>
<tr>
<td>Repository</td>
<td>no</td>
</tr>
<tr>
<td>Gateway</td>
<td>no</td>
</tr>
<tr>
<td>Cluster controller</td>
<td>no</td>
</tr>
<tr>
<td>Cache server</td>
<td>no</td>
</tr>
<tr>
<td>Server process</td>
<td>Requires database driver?</td>
</tr>
<tr>
<td>------------------</td>
<td>---------------------------</td>
</tr>
<tr>
<td>Search &amp; Browse</td>
<td>no</td>
</tr>
<tr>
<td>File store</td>
<td>no</td>
</tr>
</tbody>
</table>

Reinstall and Configure Worker Node

You might need to reinstall one of your Tableau worker nodes. To do so, follow one of these procedures. The specific steps you take depend on whether or not the worker you are reinstalling has data engine or repository components on it and whether or not these are duplicated on any other node in the installation.

**Note:** Reinstalling multiple workers at the same time could lead to data loss.

Use the following procedure to help you reinstall and configure a worker node that is hosting the only file store or repository in the distributed installation. Every Tableau Server installation requires at least one file store and one repository. If you are reinstalling the worker node that hosts either of these processes, you must first add the process to a second node.

To reinstall the worker node hosting the file store or repository instance

1. Create a full backup of Tableau Server. For more information, see Back Up Tableau Server Data.

2. Stop Tableau Server on the primary by selecting **Tableau Server 10.5 > Stop Tableau Server** on the Windows Start menu, or by running the `tabadmin stop` command from the command line.

3. On the Start menu, select **Tableau Server 10.5 > Configure Tableau Server**.

4. In the Configuration Utility:
• On the **General** tab, enter your password.
• On the **Servers** tab, add the file store and/or repository components that the worker is hosting to another worker or to the primary, and then save your changes.
  For example, if the worker you are reinstalling currently hosts the file store, add the file store to another node.

5. **Start the primary Tableau Server node** so that synchronization completes between the existing file store or repository on the worker you will be reinstalling and the newly added instances of those processes.

6. Open the Status page in Tableau Server and check on the components you added:
   • If you added a file store, wait until the new file store status no longer says "Syncing".
   • If you added a repository, wait until the new repository status says "Passive".

7. **Stop Tableau Server.**

8. If you are removing a node that hosts file store, **decommission** the file store you are removing:

   From the Windows command line, in the `C:\Program Files\Tableau\Tableau Server\10.5\bin` directory, run:

   ```
   tabadmin decommission <node>
   ```

   where `<node>` is the name or ip address of the node you are going to remove, as it appears in the list of servers on the **Servers** tab of the Configuration utility.

9. In the Configuration Utility:
   • On the **General** tab, enter your password.
   • On the **Servers** tab, select the worker you want to reinstall and then click **Delete**.
   • Save your changes.
10. Start Tableau Server and verify that everything is working as expected.

11. On the worker:
   - Uninstall the Tableau Server worker software from Windows Control Panel.
   - Delete (or rename) the following folders: C:\Program Files\Tableau and C:\ProgramData\Tableau. \ProgramData is a hidden folder so may not be visible.
   - Install the updated worker software.

12. On the Tableau Server primary, stop Tableau Server, add the worker back into the configuration, and then save the changes.

   **Note:** The file store and repository need to remain on at least one node while you are re-adding the worker.


Use the following procedure to help you reinstall and configure a Tableau worker that is either not hosting a file store or repository, or is hosting a component but there is an additional node that is hosting the same component.

To reinstall and configure the worker node that is either not hosting file store or is hosting one that is also on another node

1. Create a full backup of Tableau Server.

2. Stop Tableau Server on the primary by selecting **Tableau Server 10.5 > Stop Tableau Server** on the Start menu or by running the tabadmin stop command at a command prompt.

3. If you are removing a node that includes a file store, decommission the file store on that node:
From the Windows command line, in the C:\Program Files\Tableau\Tableau Server\10.5\bin directory, run:

```
tabadmin decommission <node>
```

where `<node>` is the name or ip address of the node you are going to remove, as it appears in the list of servers on the Servers tab of the Configuration utility.

4. Open the configuration utility by selecting Tableau Server 10.5 > Configure Tableau Server on the Start menu.

5. In the Configuration Utility:
   - On the General tab, enter your password.
   - On the Servers tab, select the worker you want to reinstall and then click Delete.
   - Save your changes.

6. Start Tableau Server and verify that everything is working as expected.

7. On the worker:
   - Uninstall the Tableau Server Worker software from Control Panel.
   - Delete (or rename) the following folders: C:\Program Files\Tableau and C:\ProgramData\Tableau. \ProgramData is a hidden folder so may not be visible.
   - Install the updated worker software.

8. On the primary node, stop Tableau Server, use the configuration utility to add the worker back into the configuration, and then save the configuration.

   **Note:** The file store and repository need to remain on at least one node while you are re-adding the worker.

Maintain a Distributed Environment

After you set up a primary and one or more worker servers for a distributed installation, you can perform all subsequent configuration and updates from the primary server, using the command line tools and configuration utility on the primary server. Updates will be pushed to the workers automatically.

When you installed worker servers, you specified the primary's IPv4 address or computer name. If that IP address or computer name changes, you will need to re-install the worker servers.

You can monitor the status of the Tableau Server cluster on the server Maintenance page. See Server Maintenance to learn more about maintaining the server.

Move the Repository Process

If you need to delete a worker node from your Tableau Server configuration and that worker is hosting the only instance of the repository, you must move the process to another computer before deleting the node. There must always be at least one active instance of the repository, so you cannot remove an instance if it is the only instance.
Beginning with version 10.0, you cannot have more than one instance of the repository unless you have at least three nodes. The steps below assume you have at least three nodes in your Tableau Server installation. If you need to move a repository in a two-node environment, you need to follow a set of manual steps using tabadmin. For more information, see Move the Repository Process in a Two-Node Installation.

**Note:** If you are also moving a file store process, you can move the repository at the same time. See Move the File Store Process.

1. Create a full backup of Tableau Server. For more information, see Back Up Tableau Server Data.

2. If you haven’t done so already, stop Tableau Server and run the Tableau Server Configuration utility (**Start > Tableau Server 10.5 > Configure Tableau Server**) on the primary Tableau Server node.

3. On the **Servers** tab, select the computer (IP address or computer name) onto which you want to move the repository and click **Edit**. It can be another worker or the **primary** (**This Computer (Primary)**).

4. In the **Edit Tableau Server** dialog box, select the **Repository** check box and click **OK** to close the dialog box.

5. Click **OK** in the Tableau Server Configuration utility to save your changes and close the utility.

6. **Start the primary Tableau Server node** so that synchronization completes between the existing repository and the newly added repository.

7. Open the Status page in Tableau Server and wait until the new repository status no longer says "Setting up". When the repository status is "Passive" the synchronization is complete.

8. **Stop the server** and open the Tableau Server Configuration utility.
9. On the **Servers** tab, highlight the computer from which you are removing the process and click **Edit**.

10. Remove the process you are moving: clear the **Repository** check box and click **OK**.

11. Click **OK** again to save your changes and close the utility.

12. **Start the primary server** so that the changes can take effect.

If you are performing this procedure as part of deleting a worker node from the Tableau Server configuration (as described in Remove a Worker Node) stop Tableau Server again before proceeding.

**Move the File Store Process**

If you need to delete a worker node from your Tableau Server configuration and that worker is hosting the only instance of the file store (which handle extracts), you must first move the processes to another computer before deleting the node. There must always be at least one instance of the file store process, so you cannot remove an instance if it is the only instance.

1. Create a full backup of Tableau Server. For more information, see Back Up Tableau Server Data.

2. If you haven’t done so already, **stop the primary Tableau Server node** and run the Tableau Server Configuration utility (**Start > Tableau Server 10.5 > Configure Tableau Server**) on the primary Tableau Server node.

3. On the **Servers** tab, highlight the computer (IP address or computer name) onto which you want to move the processes and click **Edit**. It can be another worker or the primary (**This Computer (Primary)**).

4. In the **Edit Tableau Server** dialog box, select **File Store** and click **OK** to close the dialog box.
Note: When you install the file store on a node, the data engine process is also installed. Selecting the File Store box also selects the Data Engine box.

5. Click OK in the Tableau Server Configuration utility to save your changes and close the utility.

6. Start the primary Tableau Server node so that the changes can take effect.

7. Open the Status page in Tableau Server and wait until the new file store status no longer says "Synchronizing".

8. Stop the server.

9. Decommission the original file store:

   From the Windows command line, in the C:\Program Files\Tableau\Tableau Server\10.5\bin directory, run:

   tabadmin decommission <node>

   where node> is the name or ip address of the node you are going to remove, as it appears in the list of servers on the Servers tab of the Configuration utility.

10. Open the Tableau Server Configuration utility and on the Servers tab, highlight the computer from which you are removing the process and click Edit.

11. Remove the process you are moving: clear the File Store box and click OK. The Data Engine check box will be cleared automatically.

12. Click OK again to save your changes and close the utility.

13. Start the primary server so that the changes can take effect.

If you are performing this procedure as part of deleting a worker node from the Tableau Server configuration (as described in Remove a Worker Node) stop Tableau Server again before proceeding.
Remove a Worker Node

To delete a worker from your Tableau Server configuration:

1. Stop the server on the primary Tableau Server.

2. On the primary server, open the configuration utility by selecting Tableau Server <version> > Configure Tableau Server on the Start menu.

3. In the configuration utility, select the Servers tab.

4. If the worker is hosting the file store or the repository, move those processes onto another machine before continuing. See Move the File Store Process for steps.

5. Next, highlight the worker and click Delete.

6. Click OK.

7. Start the server.

Update Configuration for a New IP Address

You can and should use hostnames (computer names) of your server computers when configuring Tableau Server. Doing this makes it relatively easy to update the server configuration if an IP address changes, and using hostnames is the only way Tableau Server supports IPv6 networks.

If you use IP addresses instead of hostnames when initially configuring Tableau Server, the process to update the configuration when an IP address changes is significantly more complicated. Tableau Server uses IP addresses for communications between processes and nodes. As a result, if an IP address changes, Tableau Server will not function properly until you update the server configuration. This might happen if a VM is down for a significant amount of time (longer that the DNS record time to live) and the network environment uses DHCP.
Updating a Tableau Server configuration that was originally configured using hostnames

To update the configuration for a new primary node IP address if you configured using hostnames:

- On each worker node, restart Tableau Server.

To update the configuration for a new worker node IP address if you configured using hostnames:

1. On the primary Tableau Server computer, open a command prompt as an administrator.

2. Navigate to the server \bin directory. For example, in a default installation, type the following:
   
   `cd "C:\Program Files\Tableau\Tableau Server\<version>\bin"
   
3. Stop the server:
   
   `tabadmin stop`

   **Note:** If the IP address on the primary node has changed, the `tabadmin stop` command will not work because the worker nodes will not accept connections from the new IP address. If you cannot stop Tableau Server, restart the worker nodes and then try again to stop the server.

4. Update the configuration:

   `tabadmin config`

5. Start the server:

   `tabadmin start`
Move the Repository Process in a Two-Node Installation

The Tableau Server repository is installed on the primary node by default. If your distributed installation has at least three nodes total (one primary and two workers), you can move the repository to another node (see Move the Repository Process), or add a second repository instance to another node (see Install and Configure Worker Nodes).

If you have a two-node Tableau Server cluster, moving the repository from the primary node to the worker node requires you to use the tabadmin utility to issue a series of commands. This move process between the primary node and a single worker is only supported from the command line. You cannot move a repository using the Tableau Server configuration utility.

Before you start: For this procedure to work, you must specify each node using the identifier that Tableau Server uses for the node. You can find these on the Status page, or in the output of the `tabadmin status -v` command. The names or IP addresses you use in the commands below must match what is shown on the Status page or the command output. Names are case-sensitive.

**Important:** To successfully complete this procedure, you must follow the steps in order and as described. If you do not, you could lose your Tableau Server data. You should only attempt this if you are knowledgeable about the tabadmin utility and comfortable using it to configure Tableau Server.

1. Create a full backup of Tableau Server. Keep this backup in a safe location that is not on one of your server computers. For more information, see Back Up Tableau Server Data.

2. Open a command prompt as administrator and navigate to the Tableau Server `\bin` directory. By default this is located here:

   `C:\Program Files\Tableau\Tableau Server\10.5\bin`

3. Set the primary as one host of the repository:
tabadmin set pgsql0.host [primary_ip or primary_name]

**Important:** Be sure you specify the primary node. If you mistakenly provide an IP address or name that does not belong to the primary, Tableau Server will no longer recognize your existing repository data and you cannot fix this without restoring your backup, created in step 1 above.

4. Set the worker as another host of the repository:
   
   tabadmin set pgsql1.host [worker_ip or worker_name]

5. Restart Tableau Server:
   
   tabadmin restart

   The `restart` command runs three separate commands: `stop`, `configure`, and `start`. During the start step, a warning message will display about an invalid value in `tabsvc.yml` and a requirement of a minimum of three nodes for multiple repositories:

   ```
   *** WARNING: Invalid value(s) found in: 'C:\ProgramData\Tableau\Tableau Server\config\tabsvc.yml'
   A value of 'server_name' instances for 'pgsql1.host' is not recommended. Tableau Server requires a minimum of three nodes to support multiple repositories and repository failover. Either add a third node or remove one instance of the repository.
   
   You can ignore this message while following this procedure.
   ```

6. When Tableau Server starts, the repository on the primary will begin synchronizing with the new repository, copying all data to that second repository. Wait for the repository sync to complete, so all data in the repository on the primary node is also in the repository on the worker.
You can check the server status page to see if the repositories are syncing. To verify that the sync process is complete, at a command prompt, run `tabadmin status -v`. The worker will be listed by IP address and should include a line for the repository:

'Tableau Server Repository' (process ID) is running (Passive Repository)

If the repository is listed as "running (Passive Repository)" then the sync process is complete.

7. Reset the default for the `pgsql<n>` values. These are only used when there are two repository instances:

   `tabadmin set pgsql0.host -d`
   `tabadmin set pgsql1.host -d`

8. Set the worker as the only host of the repository:

   `tabadmin set pgsql.host [worker_ip or worker_name]`

   **Important:** Be sure you specify the worker node the primary repository synchronized to. If you do not, Tableau Server will not recognize the repository and you cannot fix this without restoring your backup, created in step 1 above.

9. Restart Tableau Server:

   `tabadmin restart`

**High Availability**

A highly available installation of Tableau Server is a distributed installation that is designed to maximize the availability of Tableau Server.
If you’re configuring a Tableau Server installation for high availability (HA), the steps you perform are all designed to build in redundancy, thus reducing your potential downtime. The four areas of Tableau Server that require redundancy are the file store, repository (pgsql), and gateway processes, along with the primary node, which runs the server’s licensing component. Because there must always be one active of the repository process, configuring the cluster is a multi-phased procedure that requires the primary Tableau Server to be stopped and restarted at certain points so that settings can take effect. For exact steps, see Configure for Failover and Multiple Gateways and Use a Backup Primary. See Distributed Requirements as well.

The minimum supported configuration for high availability is a three-node system. This includes a primary server node to run licensing, and two worker nodes to host the main processes. You can increase reliability of the system by adding a fourth computer to serve as a backup primary. If you run a gateway process on all nodes, it also makes sense to use a load balancer for the gateways.

A Single Server System

After you install the primary Tableau Server, it is running at least one instance of all server processes. This is the most basic configuration of Tableau Server. It has no redundancy.
Here’s what the Process Status table on the Server Status page typically looks like for a single-server system (not all processes shown above appear on the Status page):
To build in redundancy, you need to add additional servers to host copies of the repository and data engine/file store processes. In addition, to reduce the system’s vulnerability, you can run multiple gateways, and the primary should be isolated on its own node, ideally running as few of the server processes as possible. The fewest number of computers required to achieve this is three (see A Three-Node System).

A Three-Node System

A three-node system helps you reduce the primary’s vulnerability:
This configuration would look like the following Process Status table on the Server Status page.
In a three-node cluster, the Data Engine and Repository processes have been moved from the primary to a worker, and the primary is only running the Gateway and Search & Browse processes. In this configuration, if your active worker fails, the passive worker automatically becomes active. Exactly how to create this three-node cluster, including how to add the workers and remove the processes from the primary, is described in Configure for Failover and Multiple Gateways. (Licensing functionality is integral to the primary and cannot be removed, so it is not displayed on the Status page. Cluster Controller and Coordination Service are installed on all nodes as part of the "base install" and are not configurable. Coordination Service does not show on the Status page and Cluster Controller only displays if there are two or more nodes in the cluster.)
There are still two things you can do to improve this three-node cluster: 1) add a load balancer to interface with the three active gateways, and 2) create a backup to address the single point of failure: the primary. See the topics below for details.

Add a Load Balancer

At this point, all three nodes have gateways, which are used to route requests to available server processes. Unlike the repository process, there aren't active and standby gateways. All gateways are active. To further reduce your cluster's potential for downtime, you should configure a load balancer.

Add a Backup Primary

Adding a backup primary provides a safeguard for your system. The backup primary is an additional server added to the system to be ready if your primary fails. While it is not an active server, after you complete the first set of steps in Use a Backup Primary, it is ready to be activated. While the backup primary needs to be licensed during installation, it does not count as one of the three environments allowable under the Tableau EULA.

Here's what the system looks like with a backup primary:
The Process Status table for the configuration shown above looks the same as for a three-node system. If the primary fails and you perform the steps for the backup primary to take over, your system is back online using the new primary:
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 do 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 primary 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 ProgramData\Tableau\Tableau Server-\data\tabsvc\logs\licensing folder. For more information about licensed processes, see Licensed processes.

Quick Start: Configuring Failover & Highly Available Gateways

Extracts and repository data can change rapidly and even regular backups may not help you fully recover from a system failure. Another vulnerability is having a single entry point, or gateway, for your Tableau Server cluster. To help with this, distributed Tableau Server deployments provide real-time content replication and failover support, as well as the ability to run multiple gateways.

1. Install the Servers

Install Tableau Server on the primary computer. After Setup, stop the server and run Tableau Worker Setup on the two additional computers that will provide failover support. During Worker Setup, provide the primary’s IPv4 address or name.

To stop or start the server, at a command prompt, go to the Tableau Server bin folder and type `tabadmin stop` or `tabadmin start`.

Stop the primary server and open its Configuration utility.
2 Set Up Email Alerts

After you add the second worker and with the Configuration utility still open, click the SMTP Setup tab. Enter the name of your SMTP server—and a username and password if it’s required by your SMTP server.

Next, enter the email account that will send an alert if there’s a system failure, and the account(s) that will receive it.

Click the Alerts and Subscriptions tab in the Configuration utility and select Send email alerts for server component up, down, and failover events:
When you test, your email account will receive messages about the services.

Click OK and start Tableau Server.

3 Configure the Distributed System

1. On the Servers tab, click Add to add a worker server. Enter its computer name or IPv4 address. Enter 1 for each process. Select Repository, Gateway, and Search & Browse. Click OK:

![Tableau Server Configuration](image)

2. Click Add to add a second worker server. Enter its IPv4 address or computer name. Enter 1 for every process). Leave File Store and Repository cleared but select Gateway. Click OK.

3. Click OK to close the Configuration utility, then start Tableau Server on the primary server so your changes can take effect.

4. Stop the primary server and open the Configuration utility.

5. On the Servers tab, select the second worker and click Edit. Select the File Store and the Repository check boxes. Click OK, then OK again to close the Configuration utility. Start Tableau Server.
6. Still on the **Servers** tab, select **This Computer (Primary)** and click **Edit**. Set every process to 0, clear the **File Store** and **Repository** check boxes but keep **Gateway** selected. Click **OK**.

4 Load Balance the Gateways

You can optionally use a load balancer to ensure the cluster’s availability in the event of gateway failure, and to distribute the cluster’s workload.

In your load balancer, enter the IP address for each computer that’s running a gateway process (the primary and the two workers), and configure the load balancing method, such as Fastest or Round Robin

**Quick Start: Creating a Backup Primary**

This Quick Start describes how to create a backup of your primary Tableau Server computer so that if your primary fails, it will take just a few steps to bring your backup primary online.

Before beginning, make sure you have configured your environment for failover and highly available gateways, using the Quick Start: Configuring Failover & Highly Available Gateways as your guide. You should have two worker nodes and a primary Tableau Server. To help ensure a smooth transition for your Tableau Server users, assign the same common name to both your current and backup primary servers.
Configuring Primary Failover

1 Configure the Primary

Stop the server on your primary Tableau Server computer, then run the following command from the Tableau Server bin directory:

`tabadmin failoverprimary --primary "<computer1>,<computer2>"`
computer1 is the current primary node computer name or IPv4 address. computer2 is the backup primary node computer name or IPv4 address.

2 Copy the Primary Config and Keystore

Copy configuration and keystore information from the primary node to the backup primary. The configuration file is located in a hidden folder: C:\ProgramData\Tableau\Tableau Server\config\tabsvc.yml. The keystore is the tabsvc folder and everything it contains, located in ProgramData\Tableau\Tableau Server\config\tabsvc. Copy both the tabsvc.yml file and the tabsvc folder to a temporary location on the backup primary node. Edit the copy of tabsvc.yml and replace all occurrences of the IPv4 address or computer name for the primary node with the IPv4 address or computer name for the backup primary node.

3 Install & Disable the Backup Primary

Install Tableau Server on your backup primary. After Setup completes, open a command prompt on the backup primary and stop the server. Next, run the following commands:

```
tabadmin autostart off
tabadmin config
```

Before you begin the next section, power down your primary to simulate a system failure.

After the Primary Fails

4 Configure the Backup Primary

On your backup primary, use the tabsvc.yml file you edited in step 2 to overwrite the locally installed tabsvc.yml and the tabsvc folder you copied to overwrite the locally installed ProgramData\Tableau\Tableau Server\config\tabsvc folder on the backup primary node. (If web data connectors were imported to the primary server, copy them to the primary backup.) Next, open a command prompt on your backup primary and run the following commands from the backup primary’s Tableau Server bin directory:

```
tabadmin config
```
tabadmin failoverprimary --primary "<computer2>,<computer1>"

Where `<computer2>` is the computer name or IPv4 address of your backup primary (soon to be your active primary) and `<computer1>` is the computer name or IPv4 address for your former primary (soon to be your backup).

5 Start the Backup Primary

Run the following command:

```
    tabadmin autostart on
```

```
    tabadmin config
```

Then start the computer. Your backup primary is now your primary.

6 View Status

Sign in to Tableau Server on your new primary and view the status of your distributed system on the Status page. In the first row of the Status table you’ll see the computer name or IP address of your new primary server.

**Configure for Failover and Multiple Gateways**

Do the following to configure a three-computer cluster that provides multiple gateways and failover support. In most cases, running multiple gateways makes sense only if you plan to also use a load balancer.

1. **Install Tableau Server** on your primary computer.

2. After Setup completes, check the Status page. All the processes should have a green “active” status:
3. **Stop the server** on the primary computer.

4. Run **Tableau Worker Setup** on the two additional computers or VMs that will provide failover and extra gateway support. During Worker Setup, you will need to provide the computer name (recommended) or IPv4 addresses of the primary Tableau Server. If you enter multiple IPv4 addresses, separate each with a comma. The computer name that you specify here should be the Domain Name System (DNS) host name.
Note: The primary computer must have a static IP address assigned to it, even if you are using the primary’s computer name to identify it. For more information, see Update Configuration for a New IP Address.

5. With the primary server still stopped, start the Tableau Server Configuration utility: Start > All Programs > Tableau Server > Configure Tableau Server. On the General tab enter the Run As account password.

6. On the Servers tab, click Add to add a worker.

   The Add button is not available if you are configuring a server that is licensed with a Tableau Server—Single-Machine Core license.

7. Enter the IPv4 address or computer name of the worker, select File Store check box (Data Engine will be automatically selected), and select the Repository check box.
If you want the worker to run other server processes, enter the number of instances you want to run, such as 1 or 2. Click OK to close the Add Tableau Server dialog box.

**Note:** If you install Tableau Server on a two-node cluster, a message displays to let you know that you are limited to one instance of the repository, and that high availability and failover are not available in a two-node configuration. You can add a third node but are not required to do so. In a two-node cluster, if one of the two nodes goes down, Tableau Server may not function correctly.

8. 

9. Click **Add** on the **Servers** tab to add another worker.

10. Enter the IPv4 address or computer name of the second worker, enter at least 1 for every process. Select **Gateway**. Select **Search & Browse**.
Click **OK** to close the Add Tableau Server dialog box and click **OK** to save the configuration and close the Configuration utility.

11. **Start the server** on the primary computer.

12. **Important:** Allow several minutes for the server’s synchronization processes to copy data. This can take anywhere from 5 minutes to 15 minutes (or even much longer) depending on the size of your installation and the number of extracts.

13. Open the Status page in Tableau Server and check on the components you added:

   - If you added a file store, wait until the new file store status no longer says "Syncing".
   - If you added a repository, wait until the new repository status says "Passive”.

14. After you’ve confirmed that the synchronization is complete, **stop the server** on the primary.

15. Open the Configuration utility. On the **General** tab enter the Run As account password, then click the **Servers** tab, select **This Computer (Primary)**, and click **Edit**.

16. In the Edit Tableau Server dialog box, clear the **File Store** and the **Repository** check
boxes. Keep **Gateway** selected. If you want the primary Tableau Server to run nothing but the gateway process (Apache), you can remove the remaining server processes from the primary by entering 0 in each text box.

With a core-based license, the gateway and search & browse processes consume no cores. Configuring the primary Tableau Server to run nothing but the gateway and search & browse is a useful strategy if, for example, you have a 16-core server license and two 8-core workers. You can run three nodes (the primary plus two workers) and only the worker nodes are consuming cores.

![Edit Tableau Server](image)

Click **OK**.

17. On the **Servers** tab, select the first worker, click **Edit**, and select the **Gateway** check
box. Leave the other settings unchanged. Click OK.

18. Still on the Servers tab, select the second worker and click Edit.

19. Select File Store check box and select the Repository check box.

20. Click OK.

The Servers tab should now look similar to this:
21. You can also set up email alerts so that you’re notified of server failures or changes in status for your data engine and repository processes. To do this, click the **Alerts and Subscriptions** tab in the Configuration utility and follow the steps in To configure email alerts for system failures.

22. Click **OK** to close the Configuration utility.

23. If you are removing file store (step 14 above), a message appears to let you know that the file store was not decommissioned, and asking if you want to decommission it. Click **Yes** to decommission the file store.

24. **Start the server** on the primary computer (it may take a few minutes for your changes to take effect). Your system is now configured to provide failover support for the repository process. It is also configured for multiple gateways. You can now use a load balancer to ensure the cluster’s availability in the event of a gateway failure—and to distribute the cluster’s workload.

The Status page should look similar to this:
Configure Tableau Server for High Availability with Coordination Service-Only Nodes

As a part of the Tableau Server installation, a Coordination Service process is installed on each server node. Coordination Service is a service built on Apache ZooKeeper, that coordinates activities on the server. 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 the nodes that are running other server processes. This procedure explains how to do this.
To run Tableau Server with Coordination Service-only nodes

1. Install Tableau Server on the primary computer (primary node).

2. Install Tableau Server worker software on additional computers.

   You need at least three nodes to run Coordination Service, plus the nodes on which you want to run Tableau Server as part of your distributed installation. In the example below, a total of six nodes are used.

3. On the primary node, run the Configuration utility, and add the nodes on which you installed the worker software.

4. In the Configuration utility, edit each server that will run Tableau Server, and specify the processes that should be installed. For more information on how to configure a distributed installation, see Install and Configure Worker Nodes.

5. In the Configuration utility, edit each server that will run only the Coordination Service process, and configure the node so it is not running any other Tableau Server processes. These nodes are considered "external" to the Tableau Server configuration and will only run the only Coordination Service.

   **Note:** The **Base Install** process is required and installs Coordination Service. You cannot remove it.

6. Close the Configuration utility.

7. On the primary node, at the command line:

   a. Configure 0 Coordination Service processes on the nodes that are running Tableau Server processes.

   b. Update the configuration on all nodes.

   c. Start Tableau Server.
Example

The following example shows how you would configure a three-node distributed installation of Tableau Server along with three nodes running just Coordination Service. If you want fail-over support in Tableau Server, you must run Coordination Service on a minimum of three nodes so there is a quorum.

1. **Install Tableau Server** on your primary computer.

   After Setup completes, check the Server Status page. All the processes should have a green “active” status.

   ![Server Status Table](image)

<table>
<thead>
<tr>
<th>Process</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gateway</td>
<td>✔</td>
</tr>
<tr>
<td>Application Server</td>
<td>✔</td>
</tr>
<tr>
<td>VertiQL Server</td>
<td>✔ ✔</td>
</tr>
<tr>
<td>Cache Server</td>
<td>✔ ✔</td>
</tr>
<tr>
<td>Search &amp; Browse</td>
<td>✔</td>
</tr>
<tr>
<td>Backgrounder</td>
<td>✔</td>
</tr>
<tr>
<td>Data Server</td>
<td>✔ ✔</td>
</tr>
<tr>
<td>Data Engine</td>
<td>✔</td>
</tr>
<tr>
<td>File Store</td>
<td>✔</td>
</tr>
<tr>
<td>Repository</td>
<td>✔</td>
</tr>
</tbody>
</table>

2. **Stop the server** on the primary computer.

3. Run **Tableau Worker Setup** on five additional computers or VMs. Two of these will be worker nodes in the installation, run Tableau Server processes, and provide failover
support. The other three will run Coordination Service.

During worker setup, you will need to provide the computer name (recommended) or IPv4 addresses of the primary Tableau Server.

![Worker Configuration](image)

**Note:** The primary computer must have a static IP address assigned to it, even if you are using the primary's computer name to identify it. For more information, see Update Configuration for a New IP Address.

4. Start the Tableau Server Configuration utility: **Start > All Programs > Tableau Server > Configure Tableau Server**. On the **General** tab, enter the Run As account password.

5. On the **Servers** tab, click **Discover** to add the five worker nodes.

6. Select the first worker node, and then click **Edit**. Select the **File Store** check box (**Data Engine** will be automatically selected), and then select the **Repository** check box.

   Click **OK** to close the Edit Tableau Server dialog box.

7. Select the second worker node, and repeat step 6.

8. For each of the next three computers:
a. Select the computer from the Servers list, click Edit, and then enter 0 for every process. Clear the options for File Store, Repository, Gateway, and Search & Browse. Base Install will be selected and you cannot change this. These nodes will run only Coordination Service.

The configuration for each of these nodes should look like this:

![Add Tableau Server dialog box](image)

b. Click OK to close the Edit Tableau Server dialog box.

9. Click OK to close the Edit Tableau Server dialog box, and then click OK to save the configuration and close the Configuration utility.

10. At a command prompt on the primary computer, remove Coordination Service from
the primary node and the two worker nodes that are running Tableau Server:

```bash
tabadmin set worker0.zookeeper.procs 0
```

```
tabadmin set worker1.zookeeper.procs 0
```

```
tabadmin set worker2.zookeeper.procs 0
```

```bash
tabadmin config
```

**Note:** You can find the number of each node from the status page. The primary node is always `worker0`.

11. **Start the server** on the primary computer.

### Add a Load Balancer

You can enhance the reliability of a Tableau Server cluster by running multiple gateways 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.

If you plan to also create a backup primary and that computer will be running a gateway process, be sure to identify that gateway to your load balancer, along with all the other gateways.

**Note:** 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.

### Guidelines

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 primary Tableau Server.

• **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 Configure a reverse proxy server.

• **Persistence:** External load balancer configuration should not include any persistence or affinity unless Active Directory (NTLM) authentication is used. If you are using Active Directory authentication, then use cookie-based persistence for NTLM negotiation requests only.

  **Note:** You can use persistence with Kerberos enabled.

Configure Tableau Server to Work with a Load Balancer

You can configure Tableau Server to work with a load balancer by performing the following steps.
1. **Stop the server.**

2. In the Tableau Server bin directory, enter the following command, where `name` is the URL that will be used to reach Tableau Server through the load balancer:

   ```bash
   tabadmin set gateway.public.host "name"
   ```

   For example, if Tableau Server is reached by entering `tableau.example.com` in a browser address bar, enter this command:

   ```bash
   tabadmin set gateway.public.host "tableau.example.com"
   ```

3. By default, Tableau assumes that the load balancer is listening on port 80 for external communications. To use a different port, enter the following command, where `port_number` is the port:

   ```bash
   tabadmin set gateway.public.port "port_number"
   ```

   For example, if your load balancer is configured for SSL and listening on port 443, enter the following command:

   ```bash
   tabadmin set gateway.public.port "443"
   ```

4. Now, enter the following command, where `server` is the IPv4 address or computer name of the load balancer:

   ```bash
   tabadmin set gateway.trusted "server"
   ```

   The value for `server` can be a comma-separated list, for example:

   ```bash
   tabadmin set gateway.trusted "10.32.139.45, 10.32.139.46, 10.32.139.47"
   ```

   or

   ```bash
   tabadmin set gateway.trusted "proxy1, proxy2, proxy3"
   ```

5. In the next command, you will provide any alternate names for the load balancer, such as

   ```bash
   tabadmin set gateway.alternate " alternate1, alternate2, alternate3"
   ```
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:

```
  tabadmin set gateway.trusted_hosts "name1, name2, name3"
```

For example:

```
  tabadmin set gateway.trusted_hosts "lb.example.com, lb, ftp.example.com, www.example.com"
```

6. Run the `config` command:

```
  tabadmin config
```

7. Start the server so the changes can take effect.

Use a Backup Primary

Before you follow the procedures in the topics below, follow the steps in Configure for Failover and Multiple Gateways. After going through those steps, you have two worker servers that are providing failover support. Each server is also running a gateway, for which a load balancer can be configured. The primary Tableau Server is running a gateway process and licensing, which is not exposed or assignable as a process. Now that you have redundancy for the file store, repository, and gateway, you need to build in redundancy for your primary Tableau Server. You do this by creating a backup of it. While the backup primary needs to be licensed during installation, it does not count as one of the three environments allowable under the Tableau EULA.

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 do 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 primary 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 ProgramData\Tableau\Tableau Server-\data\tabsvc\logs\licensing folder. For more information about licensed processes, see Licensed processes.

Removing a Backup Primary

If you have a backup primary node that you no longer need, you can easily remove it from your Tableau Server installation. To remove a backup primary, all you need to do is use Windows Control Panel to uninstall Tableau Server.

Create a Backup Primary

This procedure describes how to create a backup primary. To do this you need to configure your primary node using a failoverprimary command that uses either computer names or IP addresses. We strongly recommend using IP addresses when installing and configuring Tableau Server because this simplifies reconfiguration if any of the IP addresses for the nodes changes. For more information about how to update a Tableau Server configuration for a new IP address, see Update Configuration for a New IP Address.

1. On the primary, open a command prompt as an administrator and navigate to the Tableau Server bin directory:

   C:\Program Files\Tableau\Tableau Server\10.5\bin

2. Stop the server:
3. Issue the `failoverprimary` command.

Enter the following command, using either the computer names for the current and backup primaries (recommended) or the IPv4 addresses for the current and backup primaries. Separate multiple computer names or IPv4 addresses with a comma and no spaces:

```
tabadmin failoverprimary --primary "primary1_name,primary2_name"
```

or

```
tabadmin failoverprimary --primary "primary1_IP,primary2_IP"
```

For example, if the computer name of the current primary is `TABLEAU_SERVER` and the computer name of the backup primary is `TABLEAU_SERVER2`, you would enter the following:

```
tabadmin failoverprimary --primary "TABLEAU_SERVER,TABLEAU_SERVER2"
```

Here's a command example that uses IPv4 addresses. This example assumes that your primary (primary1_IP) has a single IPv4 address of `10.32.139.22` and your backup primary (primary2_IP) has a single IPv4 address of `10.32.139.26`:

```
tabadmin failoverprimary --primary
"10.32.139.22,10.32.139.26"
```

If the primary and backup primary have multiple IPv4 addresses, enter them all. For example:
tabadmin failoverprimary --primary
"10.32.139.22,10.32.139.23,10.32.139.26,10.32.139.27"

4. Next, copy the following from the primary and save them to a temporary location on your backup primary computer:

- The `tabsvc.yml` file, located in ProgramData\Tableau\Tableau Server\config
- The `tabsvc` folder, located in ProgramData\Tableau\Tableau Server\config

You need to use these if you switch to your backup primary (failover).

**Important:** The `tabsvc.yml` file contains server configuration settings. It gets updated when you change your configuration settings in the Tableau Server Configuration utility or using tabadmin commands. The `tabsvc` folder contains the keystore for securing server secrets. The key store and the `tabsvc.yml` configuration file get updated when you regenerate encrypted secrets using the `tabadmin regenerate_internal_tokens` command. If you change any configuration values or regenerate encrypted secrets after making copies of the `tabsvc.yml` file and `tabsvc` folder, you need to recopy these to your backup primary and edit the `tabsvc.yml` file to ensure you have the latest configurations if you need to failover.

5. On your backup primary, edit the copy of the `tabsvc.yml` file and replace all occurrences of the computer name or IP address(es) for the primary with the computer name or IP address(es) for the backup primary (the computer you’re currently on). If the primary is only running the gateway, as described in this procedure, the only line you’ll need to edit is `worker.hosts`. 

- 243 -
6. Install Tableau Server on the backup primary. Use the same product key or keys that you used for license activation, and the same Run As account and configuration settings that you used on your primary.
7. After Setup completes, stop the server on the backup primary.

8. Still on your backup primary, enter the following command to disable the automatic starting of the Tableau Server service:

   `tabadmin autostart off`

9. Commit the configuration change:

   `tabadmin config`

You’ve finished creating a backup primary. See Switch to Backup Primary for what to do if your current primary fails.

If you are working in a test environment, this would be a good time to test your configuration by powering down your current primary to simulate a system failure.

Switch to Backup Primary

If your primary node fails, and you have a backup primary configured, follow these steps to switch to your backup primary. All steps should be performed on the backup primary computer.

1. On your backup primary, locate the `tabsvc.yml` file and `tabsvc` folder you copied and edited in steps 4 and 5 of Create a Backup Primary and copy these from your temporary location to `ProgramData\Tableau\Tableau Server\config` and replace the existing `tabsvc.yml` file and `tabsvc` folder on the backup primary. You need to do this so the backup primary has the same configuration settings as the primary did.
2. If web data connectors were imported to the primary server, copy them to the following folder on the backup primary:

C:\ProgramData\Tableau\Tableau Server-\data\tabsvc\httpd\htdocs\webdataconnectors

Copy the web data connectors from the location from which they were imported to the primary server, or extract the contents of a Tableau Server .tsbak backup file and get them from there.

3. Open a command prompt as an administrator and navigate to the Tableau Server bin directory:

C:\Program Files\Tableau\Tableau Server\10.5\bin

4. Enter the following command:

    tabadmin config

5. Enter the following command, using either the computer name (recommended) or the IPv4 address(es) of your backup primary (soon to be your new primary) and of your original primary (soon to be your backup primary). Separate multiple computer names or IPv4 addresses with a comma and no spaces.

    tabadmin failoverprimary --primary "primary2_name,primary1_name"

    or

    tabadmin failoverprimary --primary "primary2_IP,primary1_IP"

For example, if the computer name of the backup primary is TABLEAU_SERVER2 and the name of the original primary is TABLEAU_SERVER, you would enter the following:
Here's an example that uses IPv4 addresses. This example assumes that your backup primary (primary2_IP) has a single IPv4 address of 10.32.139.26 and your original primary (primary1_IP) has a single IPv4 address of 10.32.139.22:

```
(tabadmin failoverprimary --primary "10.32.139.26,10.32.139.22"

If the backup primary and original primary have multiple IPv4 addresses, enter them all. For example:

```
(tabadmin failoverprimary --primary "10.32.139.26,10.32.139.27,10.32.139.22,10.32.139.23"

6. Enter the following command:

```
(tabadmin autostart on

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

```
(tabadmin config

8. **Start the server.** Your backup primary is now your primary. When you look at the Status page, you should see that the IP address or computer name for the primary has changed:

---

**Server Status**

The real-time status of processes running in Tableau Server.

<table>
<thead>
<tr>
<th>Process</th>
<th>Primary</th>
<th>Worker 1</th>
<th>Worker 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cluster Controller</td>
<td>10.32.139.26</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>
</tbody>
</table>
9. Your original primary is now your backup primary.

Automated Installation or Upgrade of Tableau Server

If you are responsible for deploying or managing Tableau Server, installation and upgrades are essential tasks. You can do these using the Setup program and following the prompts, but beginning with version 10.1 you can also take advantage of installation switches and options to install or upgrade Tableau Server without any intervention from you (no waiting for Setup program responses or Configuration utility prompts). And if you are knowledgeable about scripting or programming you can automate the install or upgrade process so it is done in a controlled and reproducible way, whether your installation is a single node or a distributed, multi-node environment.

Prerequisites for an automated install or upgrade of Tableau Server

To install Tableau Server without any input from you, you need the following:

- The Tableau Server install program (and the Tableau Server Worker install program if you have a multi-node environment), version 10.1.

- A valid product key.

- Registration information in JSON format.

  For details, see Registration Input File Options.

- User name/password for the initial administrator user.

- (Optional: If no configuration file is included, Tableau Server is installed with all defaults, including the default Windows NT AUTHORITY\NETWORK SERVICE user for the Run As User)
• Run As User information.

• Configuration file with any non-default configuration values.

For details, see Automated Installation Configuration Options.

• A script or program to do the installation (you can use the sample Python script on GitHub, or write your own)

Best practices for an automated install

To automate the installation of Tableau Server with non-default configuration, you can provide a YML file with configuration options. Details of the configuration options are listed in Automated Installation Configuration Options, and you can find sample configuration files on GitHub. When using a configuration file to specify non-default values, you should consider using one or more of these best practices:

• Install Tableau Server interactively, with the same configuration as your new server will have, then use the resulting tabsvc.yml configuration file as a starting point for the file you use during an automated install. By default the configuration file is created in C:\ProgramData\Tableau\Tableau Server\config. Changes you need to make include IP addresses and host names for the nodes.

  Note: If you create a tabsvc.yml file you need to add the config.version value (15 for version 10.1) as the first line of the file:

  ```
  config.version: 15
  ```

  If you use an existing tabsvc.yml file that already has a config.version value, you do not need to manually update it.

• Use one of the sample configuration files from GitHub as a starting point for your file: a single-node sample configuration file and a multi-node sample configuration file.

When performing an upgrade, the existing configuration file is used, so you do not need to specify non-default configuration values, except for the Run As User username and
password (if you are not using the default). We recommend using a "secrets" file instead of in the configuration file. For more information, see Server Run As User.

## Install Switches for Tableau Server

You can use these switches when installing or upgrading Tableau Server:

<table>
<thead>
<tr>
<th>Switch</th>
<th>Description</th>
<th>Comments</th>
</tr>
</thead>
<tbody>
<tr>
<td>/SILENT</td>
<td>Install/Upgrade: Run Setup in unattended, feedback mode. Dialog boxes and UI are displayed but you cannot interact with them.</td>
<td>Server does not start automatically when installed in silent mode. To start Tableau Server, use <code>tabadmin start</code>.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Note:</strong> Use either /SILENT or /VERYSILENT, not both.</td>
</tr>
<tr>
<td>/VERYSILENT</td>
<td>Install/Upgrade: Run Setup in unattended, fully silent mode. No dialog boxes or UI are displayed.</td>
<td>Server does not start automatically when installed in silent mode. To start Tableau Server, use <code>tabadmin start</code>.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Note:</strong> Use either /SILENT or /VERYSILENT, not both.</td>
</tr>
<tr>
<td>/SUPPRESSMSGBOXES</td>
<td>Install/Upgrade: Run Setup with any operating system message boxes suppressed. The default action for the suppressed message box is taken.</td>
<td>Any errors related the operating system would normally display are hidden. For example, if an install switch is mis-typed, the error would not display.</td>
</tr>
<tr>
<td>Switch</td>
<td>Description</td>
<td>Comments</td>
</tr>
<tr>
<td>--------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>/CUSTOMCONFIG</td>
<td>Install/Upgrade: Specify the location of a custom configuration file.</td>
<td>If the switch is not used, Tableau Server is installed with defaults. See Automated Installation Configuration Options for more information. Example: <code>&lt;Setup file&gt; /SILENT /ACCEPTEULA /CUSTOMCONFIG=G=&quot;path\filename&quot;</code></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Note: If used during an upgrade, non-default values are merged with existing configuration values. As a best practice, upgrade in a separate step, before changing the configuration. This helps if you need to isolate any issues.</td>
</tr>
<tr>
<td>/LOG</td>
<td>Install/Upgrade: Specify the location of the install log file.</td>
<td>If no file location is specified, the log file is written to the user's TEMP folder. The log file is overwritten on each install. Check this log file for errors after installation. Example: <code>&lt;Setup file&gt; /SILENT /ACCEPTEULA /LOG=G=&quot;path\filename&quot;</code></td>
</tr>
<tr>
<td>/FASTUNINSTALL</td>
<td>Upgrade: Upgrade without making a full</td>
<td>Uninstalls the existing version without creating a full backup, then installs the new version. This is the equivalent of select-</td>
</tr>
<tr>
<td>Switch</td>
<td>Description</td>
<td>Comments</td>
</tr>
<tr>
<td>----------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>--------------------------------------------------------------------------</td>
</tr>
<tr>
<td></td>
<td>backup of Tableau Server data.</td>
<td><strong>Without full backup</strong> when upgrading interactively. See Tableau Server Upgrade Backup Options for details.</td>
</tr>
<tr>
<td>/UNINSTALLONLY</td>
<td>Uninstall: Uninstall Tableau Server without then installing the new version.</td>
<td>Uninstalls the existing version. Does not install the new version.</td>
</tr>
<tr>
<td>/DIR</td>
<td>Install/Upgrade: Specify a non-default install location.</td>
<td>Specifies the location to install server. If not used, Tableau Server is installed to C:\Program Files\Tableau\Tableau Server-&lt;version&gt;</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Example:</strong> &lt;Setup file&gt; /VERYSILENT /ACCEPTEULA /DIR=R=&quot;D:\Tableau\Tableau Server&quot;</td>
</tr>
<tr>
<td>/ACCEPTEULA</td>
<td>Install/Upgrade: Accept the End User License Agreement (EULA). <em>This switch is required for silent install on both primary and worker nodes.</em></td>
<td>Required. If not included when using /SILENT or /VERYSILENT, Setup fails silently.</td>
</tr>
<tr>
<td>/PRIMARYIP</td>
<td>Install (distributed): On the worker node with the IP address of the</td>
<td>Required on worker nodes. Provides the worker node with the IP address of the</td>
</tr>
<tr>
<td>Switch</td>
<td>Description</td>
<td>Comments</td>
</tr>
<tr>
<td>--------</td>
<td>-------------</td>
<td>----------</td>
</tr>
<tr>
<td></td>
<td>worker nodes, specify the IP address of the primary node. This switch is used only on worker nodes and is required when installing worker nodes.</td>
<td>primary node during installation.</td>
</tr>
</tbody>
</table>

**Example:** `<Setup file>
/VERYSILENT /ACCEPTEULA
PRIMARYIP="172.1.1.1"

**Workflow for a new, automated installation**

When scripting a Tableau Server install to a new computer or computers, these are the steps you'll follow:

1. **(Distributed installations only)** Run the Worker Software Setup program on each worker node using the /VERYSILENT, /ACCEPTEULA, and /PRIMARYIP switches to specify the IP address or host name of the primary node.

2. Run the primary Tableau Server Setup program with the /SILENT or /VERYSILENT switch, and the /ACCEPTEULA switch, along with any other switches you want to use.

3. **Activate the license using tabadmin**(`tabadmin activate`).

4. **Register server**(`tabadmin register`).

5. **Start Tableau Server**(`tabadmin start`).

6. **Create the initial administrative user**(`tabcmd initialuser`).
Workflow for an automated upgrade

When scripting a Tableau Server upgrade, these are the steps you'll follow:

1. **Run the primary Tableau Server Setup program with the /SILENT or /VERYSILENT switch, and the /ACCEPTEULA switch, along with any other switches you want to use.**

   **Note:** You can automate an upgrade to version 10.5 if you are upgrading version 9.0.x or higher on a single-node installation, or version 9.3.x or higher if you are upgrading a multi-node installation. If you are upgrading other versions of Tableau Server, you need to uninstall Tableau Server from the primary machine and all worker nodes, then follow the steps for a multi-node installation.

2. **Start Tableau Server** (`tabadmin start`).

Automated Installation Configuration Options

When you install or upgrade Tableau Server interactively by running the Setup program, the Configuration utility runs as part of the process, and you have the opportunity to configure Tableau Server. When you do an automated install or upgrade, you can include a custom configuration file with any non-default configuration values you want set. If you don't specify a custom configuration file while doing an automated installation of Tableau Server, the server is installed with defaults. If you do use a custom configuration file to set some non-default values, you do not need to include any settings that use defaults.

The tables below list configuration options, organized by Tableau Server Configuration utility tab:
A best practice for creating a configuration file for automated Tableau Server installation is to use an existing configuration file as a starting point and modify it for your new installation. For more information, see Best practices for an automated install.

General tab
Server Run As User

**Important:** For security purposes, we recommend that you do not include the Run As User credentials in a custom configuration file. You can put these separately in a "secrets" file that is secured elsewhere. You can see an example of how to do this in the sample ScriptedInstaller python script on GitHub.

<table>
<thead>
<tr>
<th>Option</th>
<th>Default Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>service.runas.username</td>
<td>NT AUTHORITY\NetworkService</td>
<td>The Windows account that Tableau Server uses when it accesses resources.</td>
</tr>
<tr>
<td>service.runas.password</td>
<td>&lt;not applicable&gt;</td>
<td></td>
</tr>
</tbody>
</table>

More information:
- For detailed information on Run As User, see Run As User.
- For information about the Run As User option on the General tab, see Server Run As User.

User Authentication

<table>
<thead>
<tr>
<th>Option</th>
<th>Default Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>wgserver.authenticate</td>
<td>local</td>
<td>Specify the type of user authentication to be used:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- local will use the user-</td>
</tr>
<tr>
<td>Option</td>
<td>Default Value</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------</td>
<td>---------------</td>
<td>------------------------------------------------------------------------------</td>
</tr>
<tr>
<td></td>
<td></td>
<td>management built in to Tableau Server.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• <code>activedirectory</code></td>
</tr>
<tr>
<td></td>
<td></td>
<td>will use Active Directory to authenticate users.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>For more information, see Authentication.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Example:</td>
</tr>
<tr>
<td></td>
<td></td>
<td><code>wgserver.authenticate: activedirectory</code></td>
</tr>
</tbody>
</table>

More information:

- For detailed information on authorization, see Authentication.
- For information about the user authentication option on the General tab, see User Authentication.

Active Directory

For more information, see User Authentication.

<table>
<thead>
<tr>
<th>Option</th>
<th>Default Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>wgserver.domain.fqdn</code></td>
<td>&lt;not applicable&gt;</td>
<td>Specify the fully qualified domain name for Active Directory.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Example:</td>
</tr>
<tr>
<td></td>
<td></td>
<td><code>wgserver.domain.fqdn: myco.lan</code></td>
</tr>
<tr>
<td><code>wgserver.domain.nickname</code></td>
<td>&lt;not applicable&gt;</td>
<td>Specify the NetBIOS name for Active Directory.</td>
</tr>
</tbody>
</table>
### Option

<table>
<thead>
<tr>
<th>Option</th>
<th>Default Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Example:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>wgserver.domain.nickname: MYCO</td>
</tr>
<tr>
<td>wgserver.sspi.ntlm</td>
<td>false</td>
<td>Specify whether server should use Microsoft SSPI to automatically sign users in based on their Windows username and password.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Example:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>wgserver.sspi.ntlm: true</td>
</tr>
</tbody>
</table>

**More information:**

- For detailed information on Active Directory, see User Management in Active Directory Deployments.
- For information about the Active Directory options on the General tab, see User Authentication.

### Gateway

<table>
<thead>
<tr>
<th>Option</th>
<th>Default Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>worker.gateway.port</td>
<td>80</td>
<td>Port number for the gateway process.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Example:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>worker.gateway.port: 8080</td>
</tr>
<tr>
<td>install.firewall.gatewayhold</td>
<td>true</td>
<td>Specifies whether Tableau Server should open the port specified by work-</td>
</tr>
</tbody>
</table>

-257-
<table>
<thead>
<tr>
<th>Option</th>
<th>Default Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>er.gateway.port</td>
<td></td>
<td>in the Windows firewall. This allows HTTP access to the server by other computers on your network.</td>
</tr>
<tr>
<td>Example:</td>
<td></td>
<td>install.fire-wall.gatewayhold: false</td>
</tr>
</tbody>
</table>

More information:
- For information about the Gateway option on the General tab, see Gateway.

Sample data and users

<table>
<thead>
<tr>
<th>Option</th>
<th>Default Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>install.component.samples</td>
<td>true</td>
<td>Installs Tableau Samples.</td>
</tr>
<tr>
<td>Note: This option must be selected during initial install. You cannot go back and add samples after Tableau Server is installed.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Example:</td>
<td></td>
<td>install.component.samples: false</td>
</tr>
</tbody>
</table>

More information:
- For information about the Sample data option on the General tab, see Sample data.
Server Crash Reporter

<table>
<thead>
<tr>
<th>Option</th>
<th>Default Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>servercrashupload.enable</td>
<td>false</td>
<td>Specifies whether crash report uploading is enabled.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Example:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>servercrashupload.enable: true</td>
</tr>
<tr>
<td>servercrashupload.scheduled_</td>
<td>01:00:00 AM</td>
<td>Scheduled time for upload of crash logs, in UTC (HH:MM:SS AM/PM).</td>
</tr>
<tr>
<td>time</td>
<td></td>
<td>Example:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>servercrashupload.scheduled_time: 03:00:00 AM</td>
</tr>
</tbody>
</table>

More information:

- For information about the Crash Reporter option on the General tab, see Server Crash Reporter.

Data Connections tab

Caching

<table>
<thead>
<tr>
<th>Option</th>
<th>Default Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>vizqlserver.data_refresh</td>
<td>&lt;empty&gt;</td>
<td>• &lt;empty&gt; = refresh less often.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Cache and reuse data for as long as possible.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• &lt;n&gt; = balanced. &lt;n&gt; is the maximum number of minutes of caching.</td>
</tr>
</tbody>
</table>
### Option

<table>
<thead>
<tr>
<th>Option</th>
<th>Default Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>• 0 = refresh more often, with the cache being refreshed with each page reload.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Example:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>vizqlserver.data_refresh: 20</td>
</tr>
</tbody>
</table>

### Initial SQL

<table>
<thead>
<tr>
<th>Option</th>
<th>Default Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>vizqlserver.initialsql.disabled</td>
<td>false</td>
<td>Specify whether to ignore initial SQL statements for all data sources.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Example:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>vizqlserver.initialsql.disabled: true</td>
</tr>
</tbody>
</table>

### More information:

- For more information about options on the Data Connections tab, see Configure Data Connections.

### Servers tab

**Important:** You must be consistent with the way you reference a particular node. For example, if you specify the primary node by machine name in `worker.hosts` you need to use the machine name for every configuration value that references the primary node.
node. If you use an IP address to specify a worker node, you need to continue use the IP address for every instance of the worker node.

Worker 0 (primary)

<table>
<thead>
<tr>
<th>Option</th>
<th>Default Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>worker0.vizqlserver.procs</td>
<td>Number of cores / 4, up to a maximum of 4 n/4</td>
<td>Number of instances of VizQL Server installed on the node. Example: worker0.vizqlserver.procs: 4</td>
</tr>
<tr>
<td>worker0.vizportal.procs</td>
<td>1</td>
<td>Number of instances of Application Server installed on the node. Example: worker0.vizportal.procs: 1</td>
</tr>
<tr>
<td>worker0.backgrounder.procs</td>
<td>2 unless there are fewer than 8 cores, in which case default is 1 instance</td>
<td>Number of instances of Backgrounder installed on the node. Example: worker0.backgrounder.procs: 2</td>
</tr>
<tr>
<td>Option</td>
<td>Default Value</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------------</td>
<td>---------------</td>
<td>------------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| worker0.cacheserver.procs | 2 unless there are fewer than 8 cores, in which case default is 1 instance. | Number of instances of cache server installed on the node.  
Example: worker0.cacheserver.procs: 2 |
| worker0.dataserver.procs  | 2 unless there are fewer than 8 cores, in which case default is 1 instance. | Number of instances of data server installed on the node.  
Example: worker0.dataserver.procs: 2 |
| worker0.filestore.enabled | true          | Specifies whether the file store is on the node.  
Example: worker0.filestore.enabled: true |
| worker0.searchserver.procs | 1             | Specifies whether the Search & Browse process is installed on the node. By default Search & Browse is installed. |
Option | Default Value | Description
--- | --- | ---
| | | Example:
| | | `worker0.searchserver.enabled: 0`
| worker0.gateway.enabled | true | Specifies whether the gateway service is installed on the node.
| | | Example:
| | | `worker0.gateway.enabled: true`

Worker 1 - n (workers)

<table>
<thead>
<tr>
<th>Option</th>
<th>Default Value</th>
<th>Description</th>
</tr>
</thead>
</table>
| worker.hosts | <not applicable> | Machine name or IP address of the primary node and any worker nodes.  
| | | Example:  
| | | `worker.hosts: tabserver, 10.32.139.21, 10.32.139.6`
| worker<n>.vizqlserver.procs | Number of cores / 4, up to a maximum of 4.  
| | | `n/4` | Number of instances of VizQL Server installed on the node.  
| | | Example:  
| | | `worker1.vizqlserver.procs: 2`
<table>
<thead>
<tr>
<th>Option</th>
<th>Default Value</th>
<th>Description</th>
<th>Example</th>
</tr>
</thead>
<tbody>
<tr>
<td>worker&lt;n&gt;.vizportal.procs</td>
<td>1</td>
<td>Number of instances of Application Server installed on the node.</td>
<td>worker1.vizportal.procs: 1</td>
</tr>
<tr>
<td>worker&lt;n&gt;.backgrounder.procs</td>
<td>2 unless there are fewer than 8 cores, in which case default is 1 instance.</td>
<td>Number of instances of Backgrounder installed on the node.</td>
<td>worker2.backgrounder.procs: 2</td>
</tr>
<tr>
<td>worker&lt;n&gt;.cacheserver.procs</td>
<td>2 unless there are fewer than 8 cores, in which case default is 1 instance.</td>
<td>Number of instances of cache server installed on the node.</td>
<td>worker1.cacheserver.procs: 1</td>
</tr>
<tr>
<td>worker&lt;n&gt;.dataserver.procs</td>
<td>2 unless there are fewer than 8 cores, in which case default is 1 instance.</td>
<td>Number of instances of data server installed on the node.</td>
<td>worker1.dataserver.procs:</td>
</tr>
<tr>
<td>Option</td>
<td>Default Value</td>
<td>Description</td>
<td></td>
</tr>
<tr>
<td>-------------------------------</td>
<td>---------------</td>
<td>-----------------------------------------------------------------------------</td>
<td></td>
</tr>
<tr>
<td></td>
<td>default is 1 instance.</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td><code>work-&lt;n&gt;.filestore.enabled</code></td>
<td>true</td>
<td>Specifies whether the file store is on the node.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td><code>work-er1.filestore.enabled: false</code></td>
<td></td>
</tr>
<tr>
<td><code>work-&lt;n&gt;.searchserver.procs</code></td>
<td>1</td>
<td>Specifies whether the Search &amp; Browse process is installed on the node. By default Search &amp; Browse is installed.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td><code>work-er1.searchserver.procs: 0</code></td>
<td></td>
</tr>
<tr>
<td><code>worker&lt;n&gt;.gateway.enabled</code></td>
<td>true</td>
<td>Specifies whether the gateway service is installed on the node.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Example:</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td><code>worker2.gateway.enabled: false</code></td>
<td></td>
</tr>
</tbody>
</table>

**Note** Data engine (Hyper) is installed when you install file store, Vizportal, VizQL Server, data server, or backgrounder and cannot be installed separately. Every node
that has an instance of one of those processes will also have a single instance of data engine (Hyper) process.

More information:

- For more information about options on the Servers tab, see Servers.

Repository

<table>
<thead>
<tr>
<th>Option</th>
<th>Default Value</th>
<th>Description</th>
</tr>
</thead>
</table>
| pgsql0.host          | <not applicable>      | Machine name or IP address of the node on which the repository is installed. Example:  
|                      |                       | pgsql0.host: tabserver                                                      |
| pgsql1.host          | <not applicable>      | Machine name or IP address of the node if the repository installed on a second node. Example:  
|                      |                       | pgsql1.host: 10.32.139.6                                                    |
| pgsql.preferred_host | <not applicable>      | Machine name or IP address of the node with the preferred repository installed, if specified. This is only applicable if the cluster includes two instances of the repository. Example:  

More information:

- For more information about options on the Repository, see Tableau Server Repository.

SMTP tab

<table>
<thead>
<tr>
<th>Option</th>
<th>Default Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>svc-monitor.notification.smtp.server</td>
<td>&lt;not applicable&gt;</td>
<td>Address of SMTP server. Example: svc-monitor.notification.smtp.server: mail.example.com</td>
</tr>
<tr>
<td>svc-monitor.notification.smtp.send_account</td>
<td>&lt;not applicable&gt;</td>
<td>User name for SMTP account.</td>
</tr>
<tr>
<td>svc-monitor.notification.smtp.port</td>
<td>25</td>
<td>Port number for SMTP server.</td>
</tr>
<tr>
<td>svc-monitor.notification.smtp.password</td>
<td>&lt;not applicable&gt;</td>
<td>Password for SMTP server account. Example: svc-</td>
</tr>
<tr>
<td>Option</td>
<td>Default Value</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------------------------</td>
<td>---------------------------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td></td>
<td>mon-monitor.notification.smtp.password: password</td>
<td></td>
</tr>
<tr>
<td><strong>svc-mon-itor.notification.smtp.ssl_enabled</strong></td>
<td>false</td>
<td>Specifies whether the connection to the SMTP server is encrypted. Note: This should be left as false. Encrypted SMTP connections are not supported for alerts or subscriptions.</td>
</tr>
</tbody>
</table>
| **svc-mon-itor.notification.smtp.from_address** | <not applicable>                                 | Email address that sends alerts. If email alerts are enabled, this needs to be in a valid email address format but does not need to be a valid address on Tableau Server. Example:  
svc-mon-itor.notification.smtp.from_address: donot-reply@example.com |
<p>| <strong>svc-mon-itor.notification.smtp.target_addresses</strong> | &lt;not applicable&gt;                                 | Email address to receive alerts. If email alerts are enabled, you need to include at least one address. Separate multiple addresses with commas. |</p>
<table>
<thead>
<tr>
<th>Option</th>
<th>Default Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example:</td>
<td></td>
<td></td>
</tr>
<tr>
<td>svc-moni-tor.notification.smtp.target_addresses:</td>
<td></td>
<td><a href="mailto:iluvdata@example.com">iluvdata@example.com</a></td>
</tr>
</tbody>
</table>

More information:

- For more information about options on the SMTP Setup tab, see Configure SMTP Setup.
**Alerts and Subscriptions tab**

**Alerts and Subscriptions**

<table>
<thead>
<tr>
<th>Option</th>
<th>Default Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>subscriptions.enabled</td>
<td>false</td>
<td>Specifies if users can receive email for subscriptions to view.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Example:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>subscriptions.enabled: true</td>
</tr>
<tr>
<td>svc-monitor.notification.smtp.enabled</td>
<td>false</td>
<td>Specifies whether email alerts are sent for server health events (components</td>
</tr>
<tr>
<td></td>
<td></td>
<td>going up or down, failover occurring).</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Example:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>svc-monitor.notification.smtp.enabled: true</td>
</tr>
</tbody>
</table>

**Disk Space Monitoring**

<table>
<thead>
<tr>
<th>Option</th>
<th>Default Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>storage.monitoring.record_history_enabled</td>
<td>true</td>
<td>Specifies whether disk space usage should be saved in the repository.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Example:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>storage.monitoring.record_history_enabled: true</td>
</tr>
<tr>
<td>Option</td>
<td>Default Value</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------------------</td>
<td>---------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>history_enabled</td>
<td>false</td>
<td>Specifies whether email alerts will be sent if thresholds are crossed.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Example:</td>
</tr>
<tr>
<td>storage.monitoring.email_enabled</td>
<td>false</td>
<td>storage.monitoring.email_enabled: true</td>
</tr>
<tr>
<td>storage_monitoring.warning_percent</td>
<td>20</td>
<td>Warning threshold as a percentage of unused disk space.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Example:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>storage.monitoring.warning_percent: 30</td>
</tr>
<tr>
<td>storage_monitoring.critical_percent</td>
<td>10</td>
<td>Critical threshold as a percentage of unused disk space.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Example:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>storage_monitoring.critical_percent: 20</td>
</tr>
<tr>
<td>storage.monitoring.email_interval_min</td>
<td>60</td>
<td>Interval in minutes of disk space email alerts.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Example:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>storage.monitoring.email_interval_min: 45</td>
</tr>
</tbody>
</table>

More information:
For more information about options on the Alerts and Subscriptions tab, see Configure Server Alerts.

SSL tab

SSL requires that the SSL certificate and key pair be installed in an \SSL folder parallel to the \<version> folder where Tableau Server is installed. For example, C:\Program Files\Tableau\Tableau Server\SSL.

External Webserver SSL

<table>
<thead>
<tr>
<th>Option</th>
<th>Default Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>ssl.enabled</td>
<td>false</td>
<td>Specifies whether SSL is enabled for connections between clients and Tableau Server.</td>
</tr>
<tr>
<td>ssl.cert.file</td>
<td>&lt;not applicable&gt;</td>
<td>Fully-qualified path to the certificate file (.crt).</td>
</tr>
<tr>
<td>ssl.key.file</td>
<td>&lt;not applicable&gt;</td>
<td>Fully-qualified path to the key file (.key).</td>
</tr>
<tr>
<td>ssl.chain.file</td>
<td>&lt;not applicable&gt;</td>
<td>Fully-qualified path to the certificate chain file (optional).</td>
</tr>
<tr>
<td>ssl.client_certificate_login_required</td>
<td>false</td>
<td>Specifies whether mutual SSL and automatic login should be used. Only valid if ssl.enabled is set to true.</td>
</tr>
<tr>
<td>ssl.cacert.file</td>
<td>&lt;not applicable&gt;</td>
<td>Fully-qualified path to CA certificate file (.crt).</td>
</tr>
</tbody>
</table>

More information:

- For more information about options for configuring External SSL, see Configure External SSL.
### Internal Repository Database SSL

<table>
<thead>
<tr>
<th>Option</th>
<th>Default Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>pgsql.ssl.enabled</code></td>
<td>false</td>
<td>Specifies whether SSL is enabled for connections to the repository.</td>
</tr>
</tbody>
</table>
| `pgsql.ssl.required` | false         | • Required for all connections: if true and `pgsql.ssl.enabled` = true  
|                    |               | • Optional for client connection: if false and `pgsql.ssl.enabled` = true   
|                    |               | • Off for all connections: if false and `pgsql.ssl.enabled` = false         |

**More information:**
- For more information about options for configuring Internal SSL, see Configure Internal SSL.

### SAML tab

SAML requires that the SAML XML and certificate and key pair be installed in in an `\SAML folder parallel to the \<version> folder where Tableau Server is installed. For example, C:\Program Files\Tableau\Tableau Server\SAML.

<table>
<thead>
<tr>
<th>Option</th>
<th>Default Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>wgserver.authentication.login</code></td>
<td>&lt;not applicable&gt;</td>
<td>Specifies whether SAML is enabled. Example: <code>wgserver.authentication.login: saml</code></td>
</tr>
<tr>
<td>Option</td>
<td>Default Value</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>---------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>wgserver.saml.returnurl</td>
<td>&lt;not applicable&gt;</td>
<td>The URL that Tableau Server users will access.</td>
</tr>
<tr>
<td>wgserver.saml.entityid</td>
<td>&lt;not applicable&gt;</td>
<td>The entity ID that identifies the Tableau Server installation to the IdP.</td>
</tr>
<tr>
<td>wgserver.saml.cert.file</td>
<td>&lt;not applicable&gt;</td>
<td>Fully-qualified path to SAML certificate file (.crt).</td>
</tr>
<tr>
<td>wgserver.saml.key.file</td>
<td>&lt;not applicable&gt;</td>
<td>Fully-qualified path to SAML key file (.key).</td>
</tr>
<tr>
<td>wgserver.saml.idpmetadata.file</td>
<td>&lt;not applicable&gt;</td>
<td>Fully-qualified path to SAML XML metadata file (.xml).</td>
</tr>
<tr>
<td>wgserver.saml.protocol</td>
<td>http (if SAML is enabled)</td>
<td>The protocol used by SAML. Required if SAML is enabled.</td>
</tr>
<tr>
<td>wgserver.saml.domain</td>
<td>localhost (if SAML is enabled)</td>
<td>The SAML domain. Required if SAML is enabled.</td>
</tr>
<tr>
<td>wgserver.saml.port</td>
<td>80 (if SAML is enabled)</td>
<td></td>
</tr>
</tbody>
</table>

More information:
For more information about options on the SAML tab, see Configure Server-Wide SAML.
For more information about SAML and Tableau Server, see SAML.
For more information about SAML configuration options that you set using the tabadmin command, see tabadmin configuration options.

Kerberos tab

To use Kerberos, you need to generate .keytab file by running the configuration script and publish that file.

<table>
<thead>
<tr>
<th>Option</th>
<th>Default Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>wgserver.kerberos.enabled</td>
<td>false</td>
<td>Specifies whether Kerberos is enabled.</td>
</tr>
</tbody>
</table>

More information:
- For more information about options on the Kerberos tab, see Configure Kerberos.
- For more information about Kerberos and Tableau Server, see Kerberos.

SAP HANA tab

Requires external configuration in addition to setting configuration values.

<table>
<thead>
<tr>
<th>Option</th>
<th>Default Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>wgserver.sap_hana_sso.enabled</td>
<td>false</td>
<td>Specifies whether SAP HANA is enabled.</td>
</tr>
<tr>
<td>wgserver.sap_hana_sso.username.format</td>
<td>username</td>
<td>Specifies how Tableau Server will use for the SAP HANA username:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• username will use only the user name</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• domain_and_username will use a combination of the domain and the username</td>
</tr>
<tr>
<td>Option</td>
<td>Default Value</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------------------------</td>
<td>---------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• email will use the email address.</td>
</tr>
<tr>
<td>wgserver.sap_hana_sso.hostname</td>
<td>preserve</td>
<td>Specifies how to handle the case of the username. By default the username is left unchanged.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• preserve</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• upper</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• lower</td>
</tr>
</tbody>
</table>

More information:

- For more information about options on the SAP HANA tab, see Configure SAP HANA SSO.

**OpenID Connect tab**

<table>
<thead>
<tr>
<th>Option</th>
<th>Default Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>wgserver.authentication.login</td>
<td>&lt;not applicable&gt;</td>
<td>Specifies whether OpenID is enabled.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Example:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>wgserver.authentication.login: openid</td>
</tr>
<tr>
<td>vizportal.openid.client_id</td>
<td>&lt;not applicable&gt;</td>
<td></td>
</tr>
<tr>
<td>vizportal.openid.client_secret</td>
<td>&lt;not applicable&gt;</td>
<td></td>
</tr>
<tr>
<td>Option</td>
<td>Default Value</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>---------------</td>
<td>-------------------------------------------</td>
</tr>
<tr>
<td>vizportal.openid.config_url</td>
<td>&lt;not applicable&gt;</td>
<td>URL the IdP uses for OpenID Connect discovery.</td>
</tr>
</tbody>
</table>

More information:
- For more information about options on the OpenID Connect tab, see Configure Tableau Server for OpenID Connect.
- For more information on additional settings that are not exposed in the Configuration utility, see tabadmin configuration options.

**Registration Input File Options**

If you register Tableau Server from the command line using the tabadmin register command, you need to supply a JSON-format input file with the registration information you would have entered manually when registering interactively in the Setup program. The most common circumstance for doing this is when you automate the installation of Tableau Server using a script or program.

You can use the example registration file on GitHub or create your own, using all the fields in the table below, in any order, with the field name and its value on a single line, in double quotes, and separated by a colon. For example:

"city": "Seattle"

<table>
<thead>
<tr>
<th>Field name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>first_name</td>
<td>&lt;your first name&gt;</td>
</tr>
<tr>
<td>last_name</td>
<td>&lt;your last name&gt;</td>
</tr>
<tr>
<td>company</td>
<td>&lt;your company or organization&gt;</td>
</tr>
<tr>
<td>Field name</td>
<td>Value</td>
</tr>
<tr>
<td>------------</td>
<td>-------</td>
</tr>
<tr>
<td>email</td>
<td>&lt;your email address&gt;</td>
</tr>
<tr>
<td>phone</td>
<td>&lt;your phone number&gt;</td>
</tr>
<tr>
<td>city</td>
<td>&lt;your city&gt;</td>
</tr>
<tr>
<td>state</td>
<td>&lt;your state&gt;</td>
</tr>
<tr>
<td>country</td>
<td>&lt;your country&gt;</td>
</tr>
<tr>
<td>zip</td>
<td>&lt;your zip or postal code&gt;</td>
</tr>
<tr>
<td>title</td>
<td>&lt;your job title&gt;</td>
</tr>
<tr>
<td>department</td>
<td>&lt;your department from the following list&gt;:</td>
</tr>
<tr>
<td></td>
<td>• Accounting/Finance</td>
</tr>
<tr>
<td></td>
<td>• Engineering/Development</td>
</tr>
</tbody>
</table>
|            | • General Mgmt/Ad-
<p>|            |   ministration |
|            | • Human Resources |
|            | • Legal |
|            | • Marketing |
|            | • Operations |
|            | • Channel |
|            | • Product Management |
|            | • Purchasing/Merchandising |
|            | • Sales |
|            | • Science |
|            | • Support/Service |
|            | • Other |
| industry   | &lt;your industry from the following list&gt;: |
|            | • Aerospace &amp; Defense |
|            | • Agriculture &amp; Mining |</p>
<table>
<thead>
<tr>
<th>Field name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>• Associations &amp; Non-profits</td>
</tr>
<tr>
<td></td>
<td>• Automotive</td>
</tr>
<tr>
<td></td>
<td>• Banking &amp; Finance</td>
</tr>
<tr>
<td></td>
<td>• Business Services</td>
</tr>
<tr>
<td></td>
<td>• Construction</td>
</tr>
<tr>
<td></td>
<td>• Consumer Goods &amp; Services</td>
</tr>
<tr>
<td></td>
<td>• Education</td>
</tr>
<tr>
<td></td>
<td>• Energy &amp; Utilities</td>
</tr>
<tr>
<td></td>
<td>• Food &amp; Beverage</td>
</tr>
<tr>
<td></td>
<td>• Government</td>
</tr>
<tr>
<td></td>
<td>• Hardware</td>
</tr>
<tr>
<td></td>
<td>• Healthcare &amp; Medical</td>
</tr>
<tr>
<td></td>
<td>• Hospitality &amp; Travel</td>
</tr>
<tr>
<td></td>
<td>• Insurance</td>
</tr>
<tr>
<td></td>
<td>• Investment Services</td>
</tr>
<tr>
<td></td>
<td>• Manufacturing</td>
</tr>
<tr>
<td></td>
<td>• Media, Entertainment &amp; Publishing</td>
</tr>
<tr>
<td></td>
<td>• Pharmaceuticals &amp; Biotech</td>
</tr>
<tr>
<td></td>
<td>• Retail &amp; Distribution</td>
</tr>
<tr>
<td></td>
<td>• Software &amp; Technology</td>
</tr>
<tr>
<td></td>
<td>• Telecommunications</td>
</tr>
<tr>
<td></td>
<td>• Transportation &amp; Logistics</td>
</tr>
<tr>
<td></td>
<td>• Other</td>
</tr>
</tbody>
</table>

Upgrade
What's New in Tableau Server

What's New 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

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 tabadmin 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

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 The New Data Engine in Tableau Server.

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.

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 .

Added in Version 10.4

User
New features and enhancements related to web authoring and interacting with views on the web are now listed in What's New in Tableau for Users and Web Authoring and Tableau Desktop Feature Comparison.

To access topics that used to be located in the User Guide section of Tableau Server Help, see Using Tableau on the Web.

**Server or site administrator**

- Support for Recommended Data Sources
- Data source certification
- Enhanced commenting on views

**Support for Recommended Data Sources**

Recommended data sources show Tableau Desktop users relevant data sources published to Tableau Server. These recommendations are based on published data sources used by other users in your organization with similar usage behavior as the current user. To keep the recommendations accurate and up to date, the server checks for the following each day:

- New content - for example, new or updated data sources.
- New usage information - for example, Laura Rodriguez used the Food Catering data source and Henry Wilson used the Monthly Sales data source.

You can configure the schedule for this process under the "Recommendations Trainer" section of the General server settings page. For more information, see Server Settings (General).

**Data source certification**

To complement recommended data source functionality, certification offers a way to promote data sources through curation. Site administrators and project leaders can certify data sources that meet your organization’s security and compliance standards, or any other stand-
ards you define. For more information, see Use Certification to Help Users Find Trusted Data.

Enhanced commenting on views

Commenting has been completely redesigned to inspire conversations about data discoveries. For more information, see Comment on Views in Tableau user Help.

Comments are supported for all sites by default, but administrators can disable them for specific ones. While viewing a site, click Settings, and uncheck Allow users to comment on views.

Added in Version 10.3

The following features and enhancements are new in Tableau Server 10.3:

User

New features and enhancements related to web authoring and interacting with views on the web are now listed in What's New in Tableau for Users and Web Authoring and Tableau Desktop Feature Comparison.

Starting with version 10.3, all topics from the Tableau Server User Guide have been moved to Tableau Help, which covers exploring, analyzing, and consuming data in Tableau. This help is for people who create workbooks or data sources and publish them, and for people who want to see, edit, interact with, and share views - in Tableau Desktop, Server, or Online.

To access topics that used to be located in the User Guide section of Tableau Server Help, see Using Tableau on the Web.
We welcome your feedback on this change, as well as any feedback or ideas you might have on how to make our content better. Please use the feedback bar on the top of any page ("Was this page helpful?") to open the comment field and submit your feedback.

Server administrator

- Support for Recommended tables and joins
- Support for data-driven alerts
- Improved Initial Load Times for Workbooks

Support for Recommended tables and joins

Recommended tables and joins show Tableau Desktop users content published to Tableau Server that's popular with others at your organization, such as frequently used tables. To keep Recommendations accurate and up to date, the server regularly checks content for new usage information. You can track this process using the "Recommendations Trainer" task in the Background Tasks for Non Extracts view.

If you want to disable Recommendations, change the `tabadmin set` option,"recommendations.enabled", to false.

Support 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 an administrator, you set up data-driven alerts much like you do subscriptions. For details, see Set Up a Server for Data-Driven Alerts. For information about how users create and manage these alerts, see Send Data-Driven Alerts in Tableau User Help.

Data-driven alerts are supported for all sites by default, but administrators can disable them for specific ones. While viewing a site, click Settings, and uncheck Let users create alerts and receive alert emails under Data-Driven Alerts.
Improved Initial Load Times for Workbooks

For workbooks that connect to data extracts, Tableau Server now recomputes query results when the corresponding extract refresh tasks run. This reduces the load time for these workbooks when they are first viewed. For more information, see Configure Workbook Performance after a Scheduled Refresh.

Added in Version 10.2

The following features and enhancements are new in Tableau Server 10.2:

**Server administrator**

- Guest access per site
- Encrypted secrets storage
- OpenID Connect improvements
- Trusted authentication improvements
- SSL support for password-protected key file
- Support for SAP BW Single Sign-On
- Support for Windows Server 2016

**User**

- Support for WCAG-conformant views on the web
- Web authoring enhancements
- Legends per measure
Guest access per site

With core-based licenses, a server administrator can control Guest access per site. Guest access allows people to interact with embedded views without having to sign in.

You allow or disallow Guest access on the server Settings page. For more information, see Guest User.

Encrypted secrets storage

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.

Beginning with Tableau Server 10.2, most secrets are encrypted while at rest. See Manage Server Secrets.

OpenID Connect improvements

OpenID Connect (OIDC) support has been enhanced in this release to improve authentication scenarios with custom OIDC identity providers (IdP). Improvements include support for:

- Custom IdP claim mapping
- Ignoring JSON Web Token (JWT) validation
- Referencing static (local file) discovery document

See OpenID Connect.

Trusted authentication improvements

As of version 10.2, we’ve improved how Tableau Server stores trusted tickets in the Tableau Server repository. This change ensures that any trusted ticket content stored on Tableau Server cannot be used to impersonate users or access content protected by authentication. See How Trusted Authentication Works.
SSL support for password-protected key file

As of version 10.2, we've added support for organizations that use password-protected SSL keys. See Configure External SSL.

Support for SAP BW Single Sign-On

You can configure single sign-on (SSO) support from Tableau to SAP NetWeaver Business Warehouse (SAP BW). See Enable Single Sign-On for SAP BW.

Support for Windows Server 2016

You can install Tableau Server on Windows Server 2016. For more information about supported operating systems, see Before you install....

Support for WCAG-conformant views on the web

Embedded views that you author in Tableau Desktop, publish to Tableau Server, and embed in WCAG-conformant web pages, now meet WCAG 2.0 AA conformance and section 508 standards.

Views that are made available to users in Tableau Desktop, Tableau Reader, or Tableau Public are not yet supported for WCAG conformance. Views created or edited on the web in Tableau Server are not yet supported for WCAG conformance.

The following parts of the embedded view are WCAG-conformant:

- Sheet tabs
- The view area of the sheet
- Titles and captions
- Categorical legends
- Single Value (list) filters, which are radio buttons, and Multiple Values (list) filters,
which are check boxes.

- The View Data window

Some aspects of the Tableau interface are not yet supported, such as other filter types, parameters, and the view toolbar. Using these controls in your visualizations might make it difficult for users who use assistive technologies to fully understand your views.

Keyboard navigation, programmatic context for assistive technologies (using ARIA roles), alternative text for non-text elements and contrast standards, and authentication when signing into Tableau Server are also supported for embedded views.

For more information about creating accessible views, see Best Practices for Building WCAG-Conformant Views and Author Views for Accessibility in Tableau Help.

Web authoring enhancements

Enhancements to web edit in Tableau 10.2 include the following.

Web edit menu bar

A menu bar is now available as you edit views on the web.

```
File  Data  Worksheet  Dashboard  Analysis  Map  Format  Help
```

The Analysis menu includes the following new functionality for web editing:

- **Analysis > Aggregate Measures** - Use this command to aggregate or disaggregate data in the view. Tableau typically aggregates measures in your view, which means that it collects individual row values from your data source into a single value (which becomes a single mark) adjusted to the level of detail in your view. Disaggregating data means that Tableau will display a separate mark for every row data value in your data source. For more information, see Disaggregating Data in the Tableau Desktop online help.

- **Analysis > Stack Marks** - Stacking marks is relevant when your data view includes
numeric axes. That is, at least one measure has been placed on the Rows or Columns shelves. When marks are stacked, they are drawn cumulatively along an axis. When marks are not stacked, they are drawn independently along an axis. That is, they are overlapping. For more information, see Stack Marks in the Tableau Desktop online help.

Reference distributions

In addition to reference lines and bands, you can now create and configure reference distributions on a continuous axis. For details, see Reference Lines, Bands, Distributions, and Boxes.

Sort data in the view

To quickly sort data in your view, click the Sort buttons on the toolbar, and in the view. You can sort in ascending, or descending order.

Select a background map

If you’re editing a map view on Tableau Server, you can now select a background map.

To select a background map, select Map > Background Maps.

You can select the Tableau background map, or a WMS server or Mapbox map that the workbook author included with the workbook. You can also select None to remove a background map.

For more information, see Select Background Maps in the Tableau Help.

Legends per measure

When you include Measure Names and Measure Values in your view and drag Measure Values to color, you now have the option to create separate color legends for each measure.
in your view.

In the example below, in Edit mode, when you drag Measure Values to color in the view, by default, Tableau creates a single color legend that applies to all measures in the view.

To create separate color legends for each measure in the view, click the drop-down arrow on the Measure Values field in the Marks card and select Use Separate Legends in the context menu, Tableau creates a color legend for each measure in the view.

Note: Tableau assigns the default color palette to each new color legend. Editing the color palette for each new legend is not available for web authoring in version 10.2. If you want to use unique color palettes for each color legend, you can create the view in Tableau Desktop, edit the color palette for each legend and then publish the view to Tableau Server.
To recombine the legends, simply click on the drop-down arrow on the Measure Values field on the Marks card and select Combine Legends.

For more information about how to create separate legends for measures, see Legends in the Tableau Help.

Resize axes

You can now resize an axis in the view. To resize an axis, hover your cursor over it in the view until you see the double-sided arrow, and then click and drag the axis.
Assign colors to data items

For categorical fields, assign specific colors and custom colors (using hex code) to data items. To assign a custom color to a data item, double-click the color representing the data item, and then type the hex code for the color.
Add web page objects to dashboards

In addition to adding views to your dashboards, you can add objects, including horizontal and vertical layout containers, blank space, and now, web page objects. A web page object lets you add web-based information to your dashboard.
The result is a web page, embedded in your dashboard. For more details, see Create a Dashboard.

Set a dashboard item's exact size and position

The settings on the Layout tab give you a high degree of control over the size and position of every item in a dashboard. To set a dashboard item’s position, use the x and y settings, where a position of 0 (such as x = 0 and y = 0) is the top left corner of the dashboard. Use the w and h settings to control width and height.

For more information, see Resize and Lay Out Your Dashboard.

Added in Version 10.1

The following features and enhancements are new in Tableau Server 10.1:

User

- Web authoring enhancements
- Full-screen views
• Send subscriptions for high-priority alerts

• See user images, plus content counts in quick-search

**Server administrator**

• Automate Tableau Server install

• Let users reset passwords for local-authentication servers

**Help improvements**

• Updated server online help layout

**Web authoring enhancements**

Web authoring now supports the following capabilities:

• Show or hide headers in a view

• Add reference lines and bands to a continuous axis in a view. See Reference Lines, Bands, Distributions, and Boxes.

• View read-only device layouts for dashboards

**Full-screen views**

To expand a view, click **Full Screen** at the far right of the toolbar. To return to the view to its previous size, press Esc.

The Full Screen button appears in embedded views too. For more information, see Share Views.

**Send subscriptions for high-priority alerts**

If you’ve configured a view to contain data only when high-priority information exists, select the new subscription option, Don’t send if view is empty. For more information, see Subscribe to Views and “Manage Subscription Settings” in Manage Your Account Settings.
See user images, plus content counts in quick-search

If your organization uses Active Directory, user images automatically appear in Tableau Server. (If you update your profile image, ask a Tableau administrator to re-import your user account. For more information, see Add Users to a Site.)

Quick-search shows user images too, along with counts of workbooks, views, and data sources users own. For more information, see Search Content.

Automate Tableau Server install

Server administrators can now fully automate installation or upgrade of Tableau Server. Use the example python script on GitHub as a starting point for creating your own automated installation or upgrade. For more information, see Automated Installation or Upgrade of Tableau Server.

Let users reset passwords for local-authentication servers

Previously, if Tableau Server used local authentication, only administrators could reset passwords. Now users can reset their own passwords by clicking "Forgot password" on the sign-in page. To enable this, set the features.PasswordReset tabadmin option to true. For more information, see tabadmin configuration options.

Updated server online help layout

We’ve updated our server help!

We hope you find this format more useful than our previous versions. The most visible change is that we’ve moved the navigation to the top of the page. We’ve reorganized the
help according to roles (User, Site, and Server) so you can navigate through the content that is useful for you and the tasks you need to complete. See Get Started with Tableau Server to learn more about these roles and how we’ve organized our content.

The new format provides other improvements that are not as visible. As Tableau data nerds, we noticed that 70% of the traffic entering the Help site comes from Google. But the old content experience sent all that Google traffic to navigation-free versions of Help pages that often looked like dead-ends. Now, folks coming from Google see top- and side-nav menus that provide helpful context. This new nav structure has also motivated us to help Google users in other ways, such as combining content in longer articles to make it more likely that the first page people hit contains information relevant to them.

Tableau is also embracing mobile workflows, and this new Help layout provides a far better experience on mobile devices. The search box appears right up top rather than nested in a menu, the navigation menu requires fewer clicks, and the text is easier to read on a small screen.

We’ll continue to make more improvements in future releases. Got feedback or ideas on how to make our content better? We want it. Please use the blue feedback bar on the top of any page (Was this page helpful?) to open the comment field and submit your feedback.

Added in Version 10.0

The following features and enhancements are new in Tableau Server 10.0:

**User**

- Revision history for data sources and workbooks
- Enhancements for data source discovery
- 10.0 web authoring enhancements
Site administrator

- Site-specific SAML

Server administrator

- Administrative views for Tableau Desktop license reporting
- Administrative views for Tableau Server performance
- Subscribe other users to views and workbooks
- Customize server logos independently
- Cluster deployment
- Performance improvements
- Kerberos enhancements
- Updated server menus and navigation

Updated server menus and navigation

Top-level menus in Tableau Server have changed. For example, users who are signed in as server administrators now access individual sites by using the Manage All Sites menu.

Revision history for data sources and workbooks

As with revision history for workbooks (added in version 9.3), Tableau Server now supports revision history for data sources. Every time you publish a data source, Tableau can save a version of that data source to its revision history. In addition, workbook revisions can now be restored online in Tableau Server. For more information, see Keep Content Revisions and Maintain Content Revisions.
Enhancements for data source discovery

Published data sources can be marked as favorites, which means you can now filter and view favorite data sources.

On the Data Sources page, you can now sort data sources based on popularity, number of connected workbooks, page views based on the number of page views for workbooks, and views that are using a data source.

10.0 web authoring enhancements

In version 10.0, web authoring supports creating dashboards online, connecting to published data sources, cross-data filtering, data blending, and more.

- Create and edit dashboards
- Filter across data sources
- Blend data sources
- Rename fields
- Workbook formatting
- Data highlighting
- Show and hide cards for captions, filters, and highlighters
- Clear or duplicate sheets (for views or dashboards)
- Full support for creating and editing table calculations

Subscribe other users to views and workbooks

Content owners, project leaders, and administrators can now subscribe other users to workbooks and views in Tableau Server. For more information, see Set Up Subscriptions and Subscribe to Views.
Performance improvements

Tableau Server includes the following performance improvements to reduce load times and provide faster feedback after some workbook edits:

- To display views more quickly, the initial page load time for views has been improved.
- To display story points more quickly and provide a better experience when you navigate between them, Tableau Server now caches and pre-fetches story points.
- To provide faster feedback when you edit workbooks, the changes that you make to colors, shapes, and global transparency display more quickly.

Administrative views for Tableau Server performance

You can use two new administrative views to help optimize background tasks and view load times. 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. The **Background Task Delay** view displays how long extract refresh tasks and subscription tasks are delayed, which is the amount of time between when they are scheduled to run and when they actually run. For more information, see Background Task Delay and Performance of Views.

Customize server logos independently

You can now specify custom images for the main header, sign-in, and web authoring logos. For more information, see Change the Name or Logo and customize.

Administrative views for Tableau Desktop license reporting

Two new administrative views provide visibility into how Tableau Desktop licenses are being used in your organization, including which licenses are due for maintenance renewal or upgrade from trial versions. For more information, see Configure Desktop License Reporting.
Note: Desktop License Reporting is disabled on Tableau Server by default. See Enable and configure Desktop license reporting for details on how to enable it.

Cluster deployment

Cluster installations no longer require Active Directory domain deployment. You can now install a cluster in a Windows Workgroup. See Distributed Requirements for more information.

Site-specific SAML

Tableau Server now includes more options for authenticating users with SAML. For a Tableau Server installation with multiple sites, you can configure each site to use a separate SAML identity provider (IdP). For more information, see SAML.

Kerberos enhancements

Tableau Server now includes the following enhancements to Kerberos support:

- Kerberos authentication for Oracle data sources. See Enable Kerberos Delegation for Oracle in the Tableau Community.

- Multi-domain support for Kerberos delegation. See Kerberos delegation multi-domain configuration.

- The ability to use JDBC to connect to SAP HANA on a Mac. You can now connect to SAP HANA on a Mac using JDBC drivers and Kerberos authentication. For more information, see SAP HANA.

Added in Version 9.3

The following features and enhancements are new in Tableau Server 9.3:
Revision history for workbooks

Every time you publish a workbook, Tableau can now save a version of that workbook to its revision history. Then, if you want to revert to a previous version of the workbook, you can go to the workbook in Tableau Server, view its revision history, download the version you need, and then republish it to the server. For more information, see Quick Start: Revision History.

Support for manual-only repository failover

You can now disable automatic failover of the active PostgreSQL repository. You can disable automatic repository failover to control failover manually by using the failoverrepository command. For more information, see the clustercontroller.psql.failover option in tabadmin configuration options.
PostgreSQL database verification

You can now use the `tabadmin verify_database` command to verify that the PostgreSQL database does not contain any errors that would cause a restore to fail. For more information, see Verify the Tableau Postgres Database.

Monitoring and alerts for low disk space

Because low disk space can reduce server performance or even cause the server to stop responding, you can specify disk space thresholds and get alerts when space falls below them. Alerts continue until disk space rises above the warning threshold. You can also use a new Server Disk Space Administrative View to track recent changes in disk space. For more information, see Quick Start: Disk Space Alerts.

Content metrics and relevance in server pages

Find the right content quickly on Tableau Server and Tableau Online through metrics-based search results and sort options. List view and thumbnail view display metrics as well.

Zooming improvements for maps

Zooming in map views just got easier. You can now scroll or pinch and stretch to zoom in and out of a point on a map.

Kerberos support for PostgreSQL and Teradata connections

Tableau Server now supports Kerberos for connections to PostgreSQL and Teradata databases. For more information, see Kerberos Requirements.

On-demand connections

When you load a workbook published to Tableau Server, Tableau connects only to the data sources that are required to display your data. This means that if there are multiple tabs in a workbook, Tableau only connects to the data sources for the current tab so that you can view your data as soon as possible. For data sources that require authentication, now you only have to enter your credentials for the current tab or worksheet. As a result, when you
view a shared workbook, you can view the workbook tabs for which you have data credentials. Previously, you needed to have credentials to all the data sources in the workbook before you could view any of the workbook tabs.

**Progressive loading of dashboards**

Dashboards load progressively so that you can view and interact with your data faster. Rather than wait for the entire dashboard to load, you can start analyzing your data as soon as the first section has finished loading.

**Hardware-based defaults for single-server installations**

The Tableau Server installer detects your computer’s hardware and then creates an optimal default configuration for single-server installations. The default configuration determines the number of Tableau Server processes that will run, based on the hardware resources Tableau finds on the computer. If you run a distributed installation, only the primary computer in the cluster is configured with the new defaults. The optimal defaults are intended to serve as a good starting point for you to further optimize performance. For more information, see Primary Server Installation Defaults.

**Project Leader enhancements**

Users with the Project Leader permission can now change ownership of content, move workbooks between projects, and run extract refresh schedules. For more information, see Use Projects to Manage Content Access.

### Added in Version 9.2

The following features and enhancements were added in Tableau Server 9.2.
• Default permissions in projects

• Lock content permissions to the project

• Support for OpenID Connect

• Updated toolbars for views and web editing

• Web editing enhancements

• Find your current location in a map view

• Enhanced logging by Server Resource Manager

Default permissions on projects

Administrators and project leaders can set the permissions for a project, and set the default permissions for the workbooks and data sources in the project. As a result:

• Permission capabilities are now relevant for each content type in the project. For example, only the View, Save, and Project Leader capabilities are available for projects.

• The default permissions apply to all content published to the project. Default permissions can only be changed at the project level.

• Administrators and project leaders can give each project a unique set of default permissions.

Note: New projects in the site still start with a copy of the permissions defined for the Default project.

For more information, see Set Project Default Permissions and Lock the Project.
Lock content permissions to the project

Administrators and project leaders can lock content permissions in a project, and prevent users from changing the permissions of any content in the project. When project permissions are locked:

- The default permissions are used for all workbooks (including views) and data sources in a project.
- Users (including content owners) cannot modify permissions for individual workbooks, views, and data sources in the project.

Note: Because you can lock content permissions to the project, the Assign Permissions to Contents button has been removed for projects and workbooks.

For more information, see and Permissions in Project Hierarchies.

Support for OpenID Connect

Tableau Server now supports OpenID Connect for authentication and single sign-on (SSO). You can configure the server to redirect users to an OpenID Connect identity provider (IdP) (for example, Google), where a user signs in with the user name and password that he or she normally uses with that provider. The user is then automatically signed in to Tableau Server. For more information, see OpenID Connect.

View and authoring toolbar enhancements

The view and authoring toolbars and interfaces have been updated and a few commands have changed.
Manage and create custom views by clicking Original View or the name of the current custom view. Custom views no longer are referred to as Remember my changes.

The Export menu has been replaced with Download. To export a view as an Image, Crosstab, Data, or PDF, and to download a workbook, use the Download menu.
Web editing enhancements

The following functionality is now available as you edit views on the web:

- Drag all or part of a formula to the Data pane to create a new field.

- Right-click on a dimension in the Data pane and convert it to a measure, or right-click on a measure and convert it to a dimension:
You can also right-click to change a field’s data type, to set a default aggregation or geographic role, or to convert a measure or a date field from continuous to discrete, or from discrete to continuous.

- If you upload a workbook that uses blended data, you can see a link icon next to the field or fields in the primary data source that are being used to link the two data sources:

You can click on the link icon to activate or deactivate specific fields. When fields that can be used as linking fields are not being used, the link icon changes appearance:
- The colors that you set for the sheet tabs in Tableau Desktop also display when you edit the views in Tableau Server.

Find your current location in a map view

You can now quickly zoom to your current location in a map view.
Enhanced logging by Server Resource Manager

Administrators can now see incremental increases and decreases in memory and CPU usage for each of the processes that Server Resource Manager monitors. Entries are written into the logs for each process. Logs are located in:

<install directory>\ProgramData\Tableau\Tableau Server-\data\tabsvc\vizqlserver\Logs.

Added in Version 9.1

The following features and enhancements were added in Tableau Server 9.1.

- Scheduled synchronization of Active Directory groups
- Authentication for connected devices
- Mutual (two-way) SSL authentication
- SAML Logout
- Single Sign-on for SAP HANA
- Web data connectors

Scheduled synchronization of Active Directory groups

Server Administrators can synchronize all Active Directory groups in Tableau Server on-demand or on a scheduled basis. For more information, see Quick Start: Synchronize All Active Directory Groups on a Schedule.
Authentication for connected devices

When mobile users connect to Tableau Server, administrators can control 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 signing in after their device is successfully authenticated. For more information, see Authentication for Connected Devices.

Mutual (two-way) SSL authentication

Tableau Server now supports mutual SSL authentication between Tableau Server and clients (Tableau Desktop, web browsers, and tabcmd.exe). For more information, see Configure Mutual SSL Authentication for Tableau Server on Windows.

**Note:** When Tableau Server is configured for mutual SSL, you cannot sign into Tableau Server using a version of Tableau Desktop earlier than 9.1. If you sign in with an earlier version of Tableau Desktop and fallback authentication is not configured, an error displays: "Invalid username or password". If fallback authentication is configured, you are prompted for your user name and password. See Configure Mutual SSL Authentication for Tableau Server on Windows for details about fallback authentication.

SAML Logout

Users who sign in to Tableau Server using SAML SSO functionality can now terminate their session by signing out of server. This standards-compliant solution helps ensure SOX compliance through providing support for both Service Provider (SP) initiated logout and Identity Provider (IdP) initiated logout, with Tableau Server being the Service Provider and a third party solution like Ping Federate being the SAML Identity Provider. For more information about SAML and Tableau Server, see SAML Requirements.
Note: If you are using SAML with a version of Tableau Server prior to 9.1, see What's Changed - Things to Know Before You Upgrade.

Single Sign-On for SAP HANA

Tableau Server now supports single sign-on (SSO) for SAP HANA when SAP HANA is configured to support single sign-on (SSO). Users can sign in to their SAP HANA server and then access and publish data to Tableau Server, without having to re-enter their user name and password. For more information, see SAP HANA SSO. Note: Tableau Server requires SAP HANA driver version 1.00.9 or later to support SSO for SAP HANA.

Web data connectors

You can import web data connectors to Tableau Server. This gives you an opportunity to vet web data connectors before you make them available for users. Extracts that were created by imported web data connectors can also be refreshed on the server. You can manage web data connectors on the server using new tabadmin commands. For more information, see Web Data Connectors in Tableau Server.

Added in Version 9.0

The following features and enhancements were added in Tableau Server 9.0.

- Streamlined user interface
- Improved server management
- Improved user, group, and permission management
- Scheduled synchronization of Active Directory groups
New server processes

Faster performance

High availability

New tabadmin and tabcmd commands

Streamlined user interface

Tableau Server 9.0 features improved workflow, quick access to information, and simplified search and content management.

- A server-based service supports the rewritten web client, which is faster, more scalable, and more extensible.

- New content pages show all content by type (projects, workbooks, views, data sources) in a single place. Each page includes key information about the content, including related items and actions.

- Site navigation is streamlined. Administrators and users can switch more easily between sites and content.

- Search all content types at once using the new quick search box at the top of the page.

- View content faster by scrolling instead of paging through content.

Improved server management

Improved views, alerts, and secure communication options simplify server management

- Built-in administrative views are redesigned for improved loading and faster performance, more in-depth analytics, and an interface that’s easier to understand. For information, see Administrative Views.

- Email alerts for server processes are consolidated so that it’s easier to check server
health. For information, see Configure Server Alerts.

- The server supports enabling SSL for communication between server components and the PostgreSQL repository. For information, see Configure Internal SSL.

Improved user, group, and permission management

Manage users, groups, and permissions with a streamlined workflow.

- *Site roles* for users replace license levels and user rights. For information, see Set Users' Site Roles. Note that license levels are still used when you use the `tabcmd addusers` command to import users. For more information, see CSV Import File Guidelines.

- If a user is disabled in or is deleted from Active Directory, the user is removed from Tableau Server groups and becomes unlicensed.

- A redesigned permissions interface provides responsive visual feedback on effective and resulting permissions. You can now view and modify permissions for content in a single view. For information, see Quick Start: Permissions and Content Permissions and Ownership.

- Monitor the status of server licenses on the Licenses page, and by using the `tabadmin licenses` command. For information, see View Server Licenses.

New server processes

The following new processes help manage API requests, clusters, and high availability.

- **API Server**. This process is used when you interact with the server via REST API.

- **Cache Server**. This distributed process manages a shared query cache across the server cluster and is used by the VizQL Server, Backgrounder, and Data Server processes.

- **Cluster Controller**. This process runs on every computer in a cluster and works with
the Coordination Service to report process status and coordinate failover for high availability.

• **Coordination Service (zookeeper).** This process runs on every computer in a cluster, manages leader election when needed, and ensures that there is a quorum for making decisions during failover.

• **File Store.** This process ensures that extracts are available on all nodes of a cluster that is configured with a File Store process.

For information about the new processes, see Tableau Server Processes and Improved High Availability and Data Engine Scale.

You can see process status on the server Status page or by using the `tabadmin status --verbose` command.

The following processes from earlier versions of Tableau Server are changed or removed:

• In the Data Engine process, you are no longer limited to running only two data engine nodes per cluster. This new flexibility can improve server clusters that are used for extract-heavy scenarios.

• rSynch is removed; its functionality is now handled by File Store.

**Faster performance**

Performance enhancements make workbooks load faster, improve dashboard performance, and make interactivity more seamless with faster results and response.

• Parallel queries take advantage of the capabilities of source databases to execute more queries at the same time. Each query is faster, and independent queries start at the same time.

• Data engine vectorization speeds up the performance of extract-based workbooks.

• Parallel aggregation uses multiple cores in Tableau extracts and other file-based data
sources.

- Temp table support in the Data Server makes querying more efficient with databases that use temp tables, such as SQL Server.

- External query caching re-opens workbooks with Tableau extracts more quickly because queries are not re-run every time you open a workbook.

- Query Fusion recognizes related queries in a dashboard and combines them so there are fewer queries to execute.

- Shadow extracts create faster data source access for large text files and Excel files.

For information about optimizing Tableau Server performance, see Performance Tuning Examples.

High availability

Tableau Server provides a more secure, more robust high-availability solution with straightforward configuration and management features for administrators.

- More robust failover support:
  - An updated user experience improves cluster configuration.
  - The Cluster Controller process centralizes failure detection and response.
  - You can manually trigger Repository failover and failback using the `tabadmin` utility.
  - You can set a preferred active Repository (via a new user interface) for better support of asymmetrical hardware configurations.

- New workflows and feedback during configuration make it easier to configure a server deployment for high availability and ensure that data and server information are safe in the process.

- A new data extract storage process (File Store) provides significant speed
improvements and makes data replication more reliable.

- Tableau Server no longer restricts the number of Data Engine nodes per cluster.
- All Data Engine nodes are active and data is replicated between them, which eliminates the need for failover of the Data Engine processes.

For information, see Improved High Availability and Data Engine Scale and High Availability.

New tabadmin and tabcmd commands

The tabadmin utility has new commands, including:

- decommission
- failoverrepository
- manage_global_credentials
- recommission
- regenerate_internal_tokens

In addition, the cleanup command includes a new --reset-coordination option that extends the cleanup procedure to data maintained by the new Coordination Service.

The tabcmd utility features these improvements:

- Faster startup and execution.
- Support for Unicode.
- Support for the new --role option for the createusers, createsiteusers, and syncgroup commands.

- The following new commands:
  - deletesiteusers
  - editdomain
  - listdomains
Upgrade Tableau Server Overview

The topics in this section help you upgrade Tableau Server. The topics describe planning, testing, and actually upgrading your existing server installation. We include information about best practices, as well as 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.

Looking for Tableau Server on Linux? See Upgrade.

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.
  - What's Changed - Things to Know Before You Upgrade

  Note: As of version 10.0 there are significant changes in the Tableau Server Setup program. Read about the Tableau Server Setup changes before you upgrade to version 10.0.

- 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
  - 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.

- Understand how version compatibility might impact your installation of Tableau Server.
  - Desktop and Server Compatibility in the Tableau Help.

The following topics guide you through planning steps.

**What's Changed - Things to Know Before You Upgrade**

**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

Tableau Server configuration changes

Hyper powers the new Data Engine in Tableau Server 10.5 and introduces the changes described below. For more information, see The New Data Engine in Tableau Server.

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 Upgrading to Tableau Server 10.5.

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 Pre-Upgrade Tasks.

**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 .
What Changed in Version 10.4

Version 10.4 includes the following change you should know about before upgrading.


The following sections summarize the significant changes to Tableau Server 10.4 and provide links for additional information.

- Dashboard layouts

Dashboard layouts

The Precise dashboard spacing feature added to authoring can cause elements in existing dashboards to shift by a few pixels. This can impact layouts of dashboards that have been designed with precise layouts or floating elements. We suggest you confirm layouts of
existing dashboards during upgrade testing. For more information on the Precise dashboard spacing feature, see Precise dashboard spacing in the What's New in Tableau help for authors.

What Changed in Version 10.3

There are no changes in version 10.3 that impact upgrading.


What Changed in Version 10.2

There are no changes in version 10.2 that impact upgrading.


What Changed in Version 10.1

Version 10.1 includes some changes you should know about before upgrading.


The following sections summarize the significant changes to Tableau Server 10.1 and provide links for additional information.

- TabcmdInstaller.exe no longer installed with Tableau Server
TabcmdInstaller.exe no longer installed with Tableau Server

The Setup program for the tabcmd utility is no longer installed to the \extras folder when you install Tableau Server. The utility itself (tabcmd.exe) is still installed to the primary node when you install Tableau Server, but the separate Setup program for installing the utility on other computers is not. If you need to install the tabcmd utility on additional computers, you can download it from Tableau Software. For details, see tabcmd.

What Changed in Version 10.0

Version 10.0 includes some changes you should know about before upgrading.


The following sections summarize the significant changes to Tableau Server 10.0 and provide links for additional information.

- Tableau Server Setup changes
- Two-node installations are limited to a single instance of the repository
- Domain change from tableausoftware.com to tableau.com
- Minimum hardware requirements adjusted
- Tableau Server is no longer available as a 32-bit application
- Tableau Server no longer supports Windows Vista or Windows Server 2008
- Tableau Server no longer supports older versions of Microsoft Internet Explorer
- Tableau Server does not support Microsoft Internet Explorer 11 and higher in compatibility mode
- API Server process (wgserver) has been removed
Tableau Server Setup changes

The changes described in this section have a significant impact on the upgrade process.

Manual uninstall of previous version is no longer required

Starting with version 10.0, you can upgrade Tableau Server without first manually uninstalling your previous version (when the previous version is 64-bit 8.2 or later). When you run the setup program, the existing version of Tableau Server is recognized and is uninstalled during the upgrade process.

**Note:** If you are upgrading to version 10.0 and want, you can manually uninstall the existing version before you upgrade, following the same upgrade process as you would in versions earlier than 10.0.

Setup gives you a backup option

If you follow the new workflow and let the Setup program uninstall your existing version of Tableau Server, you are prompted during the setup process create a full backup of your Tableau installation. This backup is a safety measure and is created for use in the event of an unexpected issue during upgrade. If you already have a backup of the current state of your installation, you can skip the backup during the upgrade to save time. For more information, see Tableau Server Upgrade Backup Options.

Two-node installations are limited to a single instance of the repository

If you configure a two-node installation of Tableau Server, you are limited to a single repository. If you are upgrading from a two-node installation that has two repositories, you will be prompted to remove one instance. For more information, see Install Tableau Server on a Two-Node Cluster.

Domain change from tableau_software.com to tableau.com

As of version 10.0, the licensing server is located at tableau.com. In versions earlier than 10.0, this was located at tableau_software.com. Any firewall rules or proxy configurations that
specify tableausoftware.com must be updated for version 10.0. For more information on proxy settings, see Configuring Proxies for Tableau Server.

**Note:** Earlier versions of Tableau Server will continue to access the licensing server on the tableausoftware.com domain. If your organization is running versions of Tableau Server prior to 10.0, continue to use tableausoftware.com for proxy and firewall settings.

Minimum hardware requirements adjusted

With version 10.0, Tableau Server can be installed on a 2-core computer. (Previously, Tableau Server required at least 4 cores.) The 2-core configuration allows you to test Tableau Server on constrained hardware and is intended only for trials and prototyping. For more information, see Minimum Hardware Requirements and Recommendations for Tableau Server.

Tableau Server is no longer available as a 32-bit application

With version 10.0, Tableau Server is available only as a 64-bit application. For information about upgrading a 32-bit version of Tableau Server to version 10, see Upgrade from 32-bit to 64-bit Tableau Server.

Tableau Server no longer supports Windows Vista or Windows Server 2008

With version 10.0, Tableau Server no longer supports Microsoft Windows Vista or Windows Server 2008 (Windows Server 2008 R2 is still supported).

Tableau Server no longer supports older versions of Microsoft Internet Explorer

With version 10.0, Tableau Server no longer supports Microsoft Internet Explorer 8, 9, or 10. This change impacts customers doing an initial install of Tableau Server on Windows 8.0 or Windows Server 2012 (non-R2). Neither of these operating systems supports Internet Explorer 11.
To complete configuration of Tableau Server you must use a browser on the server computer to add an administrator account. This requires browser that is supported by Tableau Server. This means that on Windows 8.0 and Windows Server 2012 non-R2, you must use the latest version of Chrome, Firefox or Safari. You can uninstall this browser after the initial installation is complete.

**Note:** This only affects an initial installation and configuration, because no browser is required when upgrading or restoring Tableau Server.

For more information on the Microsoft policy for supporting Internet Explorer, see Microsoft Support Lifecycle.

Tableau Server does not support Microsoft Internet Explorer 11 and higher in compatibility mode

Version 10.0 of Tableau Server does not support legacy compatibility modes in Internet Explorer 11 and higher.

This change impacts you if your users view web pages that have Tableau views embedded in them and that set Internet Explorer to compatibility mode with HTML DOCTYPE values. This can be an issue with SharePoint configurations that force compatibility mode. To avoid having users view web pages that put their browser into compatibility mode, either adjust configurations so that Internet Explorer is not put into compatibility mode, or use another supported browser, such as Chrome, Firefox, or Safari.

API Server process (wgserver) has been removed

The API Server process (wgserver) has been removed from Tableau Server. The process was available but disabled by default in version 9.3. The API Server process was formerly used to support the REST API, but as of version 9.3, this functionality was moved to the Application Server process. The API Server process is now removed, even if you explicitly enabled the process in a previous version of Tableau Server. For information about changes to the REST API, see What's New in the REST API.
Note: For historical reasons, some `tabadmin` configuration options will continue to use "wgserver" in the option name, but this does not refer to the old API server.

What Changed in Version 9.3

Version 9.3 includes some changes you should know about before upgrading.

For information about what's new in Tableau Server 9.3, see the What’s New in Tableau Server topic in the Tableau Server online help.

The updates to Tableau Server 9.3 have the following impact:

New default configurations based on hardware

The Tableau Server installer detects your computer's hardware and creates an optimal default configuration for a single-server installation and for the primary server in a multi-server installation. The new default configuration determines the number of processes to run for each Tableau Server process type.

When you upgrade from a single-server or multi-server installation in which you previously accepted the default configuration, the upgrade process changes the configuration to the new hardware-based default configuration. However, if you upgrade a server where you configured a custom number of server processes, the upgrade preserves the custom configuration—both for single-server and multi-server upgrades.

If after you upgrade you want to revert to the previous default configuration, use the following table to determine the number of processes to set in the Tableau Server Configuration utility based on the number of CPU cores on the primary server:

**Number of Processes**
<table>
<thead>
<tr>
<th></th>
<th>VizQL Server</th>
<th>Data Server</th>
<th>Backgrounder</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Before 9.3</strong></td>
<td>2</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td><strong>In 9.3</strong></td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td><strong>8 Cores</strong></td>
<td>2</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td><strong>In 9.3</strong></td>
<td>2</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td><strong>12 Cores</strong></td>
<td>2</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td><strong>In 9.3</strong></td>
<td>3</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td><strong>16 Cores or more</strong></td>
<td>2</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td><strong>In 9.3</strong></td>
<td>4</td>
<td>2</td>
<td>2</td>
</tr>
</tbody>
</table>

**Note:** If the computer where you installed Tableau Server has fewer than eight CPU cores, the default configuration has not changed from running one of each process.

For more information on the defaults for 9.3, see Primary Server Installation Defaults.

For more information on setting the number of processes for Tableau Server, see Reconfigure Processes.

**High Availability Postgres Repository - faster failover**

Improvements to the failover process now mean that processes do not need to be restarted after the passive repository is made active. This means that the downtime for a repository failover is significantly reduced.

**Distributed installation - manual worker upgrades**

Due to an update in third-party software, an upgrade to version 9.3 requires manual upgrade of worker nodes. A prompt during installation of 9.3 will let you know that worker nodes cannot be upgraded automatically. For more information on upgrading, see Perform the Upgrade.

**API Server (wgserver) deprecated**

The API Server process has been deprecated. In version 9.3, the process is still available in Tableau Server, but it is disabled by default for new installations. The API Server process
was formerly used to support the REST API, but as of version 9.3, this functionality has been moved to the Application Server process. If you explicitly enabled the API Server process in a previous version of Tableau Server, the process will still be enabled in 9.3.

What Changed in Version 9.2

Version 9.2 includes some changes you should know about before upgrading.

For information about what's new in Tableau Server 9.2, see the What’s New in Tableau Server topic in the Tableau Server online help.

The updates to Tableau Server 9.2 have the following impact:

Assign Permissions to Contents setting

Because content permissions can be locked to the project, the Assign Permissions to Contents button has been removed and is no longer available for projects and workbooks. For more information, see Permissions in Project Hierarchies.

Schedules Run in Parallel by Default

When you create a schedule in Tableau Server, the schedule runs in parallel, that is, it runs on all available backgronder processes at the same time. Schedules finish more quickly when they are run in parallel, but you have the option of running schedules serially as well. For example, you may want to run a very large schedule in serial to allow other schedules to run at the same time. For more information, see Extract Refresh Schedules.

What Changed in Version 9.1

Version 9.1 includes some changes you should know about before upgrading.

For information about what’s new in Tableau Server 9.1, see the What's New in Tableau Server topic in the Tableau Server online help.
The updates to Tableau Server 9.1 have the following impact:

**SAML authentication - logout**

Starting with version 9.1, Tableau Server supports SAML logout. SAML logout is enabled by default and you can disable or enable it using the `tabadmin set wgserv-er.saml.logout.enabled false/true` command.

If your pre-9.1 Tableau Server is configured for SAML authentication, the logout functionality will not work until you reconfigure the metadata for SAML. You must re-export the SAML metadata file and re-import it into your IDP. For more information about configuring SAML metadata, see Configure Server-Wide SAML.

**Hidden fields in published data sources - unavailable for workbooks**

Starting with version 9.1, workbooks respect hidden fields in published data sources. Prior to 9.1, workbooks using hidden fields automatically exposed these fields.

If a workbook that was created prior to Tableau 9.1 used a published data source with hidden fields, the hidden fields were displayed in the workbook. Starting with Tableau 9.1, the behavior changes:

- If you are creating a new workbook that uses a published data source with hidden fields, those fields remain hidden in the workbook and cannot be used in calculations, sets, groups, and other object creation.

- If you are working with an existing workbook that uses a published data source with hidden fields, those hidden fields are displayed in red in the workbook to indicate that the fields, and therefore the views and calculations that use those fields, are invalid.

You can address this issue in one of two ways, depending on whether you want to show the fields or not:

- Show (unhide) the relevant fields in the data source, and then republish it, or

- Update the relevant workbooks to exclude the hidden fields.
For information on unhiding fields in the Data pane, see Hide or Unhide Fields in the Tableau Help.

Clickjack protection - enabled by default

Starting with version 9.1, clickjack protection is enabled by default on Tableau Server. The protection has been available for several releases, but had been off by default. For more information on clickjack protection and how it impacts embedded views, see Clickjack Protection.

**Note:** When clickjack protection is enabled, embedded views that use the embed URL copied from the browser address bar might not load. These view URLs usually contain the hash symbol (#) after the server name (for example, http://myserver-#/views/Sales/CommissionModel?:embed=y) are blocked when clickjack protection is enabled on Tableau Server.

What Changed in Version 9.0

Tableau Server 9.0 includes some changes you should know about before upgrading.


The updates to Tableau Server 9.0 have the following impact:

Customizations

Default start page

Any user-defined default start page will be reset to the Tableau Server default start page. Users will need to reset their default start page after the upgrade. For more information
about setting a default start page, search for "Manage Your Content Page and Account Settings" in the Tableau Server Help.

Custom logos

Starting with version 9.0, custom logos have changed in the following ways:

- The background for large custom logos is different based on logo location. On the navigation bar the background is black and on the sign-in screen the background is white. For more information, see Change the Name or Logo.
- The small logo option has been deprecated. There are no locations in Tableau Server where the small logo is displayed, so the option does not do anything.

Hardware Requirements (cores, RAM, and free disk space)

Beginning with version 9.0, Tableau Server will not install if your computer does not meet the minimum requirements. This is true for upgrades and new installations, and for all computers in a distributed installation. The hardware requirements are:

- **64-bit Tableau Server**—At minimum you must have 4 cores, 8 GB of RAM, and 15 GB of free disk space to install the 64-bit version of Tableau Server.

- **32-bit Tableau Server**—At minimum you must have 2 cores, 4 GB of RAM, and 15 GB of free disk space to install the 32-bit version of Tableau Server.

For more information, see Minimum Hardware Requirements and Recommendations for Tableau Server.

**Note:** If you are upgrading Tableau Server on a computer that does not meet the minimum hardware requirements, you will not be able to install Tableau Server 10.5 If you cannot upgrade 64-bit Tableau Server because of hardware requirements but your computer meets the minimum hardware requirements for 32-bit Tableau Server, you may be able to upgrade to 32-bit Tableau Server.
High availability and failover

As of version 9.0, Tableau Server no longer supports automatic failover with a two-node cluster. To get the benefit of automatic failover, you need to install Tableau Server on a minimum of three nodes. One of these can include a minimal install (the "base install" option).

The option to use an external confirmation host is no longer supported. Any installation that is configured with an external confirmation will be upgraded without that host.

When you upgrade a two-node installation that is configured for high availability (automatic failover), you are given the option to add a third node. You can do so as part of the upgrade process, or at a later time.

The Tableau Software user

Prior to Tableau Server 9.0, if you installed the sample data and users, a user named Tableau Software was created. The Tableau Software user was the owner of the sample data.

Starting with version 9.0, no Tableau Software user is created. If you install the sample data, ownership of that data is assigned to the initial user that is created (the administrator user).

Internal PostgreSQL database password regeneration

Installing Tableau Server or upgrading from a previous version regenerates the password that is used by internal Tableau Server processes for communicating with the PostgreSQL database. This password is only used by internal processes and is not accessible to server administrators or other users. For more information, see Regenerate a Password.

tabadmin restore - Doesn't automatically restart Tableau Server

Starting with version 9.0, a tabadmin restore command will not automatically start Tableau Server. If you want the server to start after doing a restore, use the --restart option. For more information, see restore.

"Remember me" option

With version 9.0 of Tableau Server, there is no Remember me option on the sign in page.
Session ID in URLs

With version 9.0 of Tableau Server, the session ID at the end of server URLs is now indicated by an "iid" parameter, :iid=<n>. For example, http://localhost/#/views/Sales2015/SalesMarginsByAreaCode?iid=1. This parameter replaces the hash symbol "# <n>" used for the session ID in 8.x versions of Tableau Server.

Changes in view URLs may impact embedded views, API calls, and trusted tickets

In Tableau Server 9.0, view URLs have changed. We recommend that you generate URLs by clicking the Share link in a view in Tableau Server 9.0, and then use the resulting URL in embedded views, API calls, or trusted tickets that you created in Tableau Server prior to version 9.0.

Note: If you use view URLs that were created by copying the URL in a browser’s address bar rather than using the URL generated by clicking the Share link, the views may not work as expected after you upgrade to version 9.0. This issue can be resolved by replacing the view URL with the Share link URL.

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** 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.

- **Minimum recommendations** are higher than 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.

Looking for Tableau Server on Linux? See [Minimum Hardware Requirements and Recommendations for Tableau Server](#).

Minimum Hardware Requirements

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. For more information on how the Setup program determines hardware, see "Determining Computer Hardware," below.

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.
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 the various factors like whether or not you will be using extracts. For more information, see Disk Space Requirements.

- 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.

**Note:** For Tableau Server 10.5, you need a minimum of 2 physical cores. If you are installing on an Amazon EC2 instance, this means 4 vCPUs. For more information, see Amazon EC2 Instances.

### 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.

<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>Install Type</td>
<td>Processor</td>
<td>CPU</td>
<td>RAM</td>
<td>Free Disk Space</td>
</tr>
<tr>
<td>--------------</td>
<td>-----------</td>
<td>-----</td>
<td>-----</td>
<td>-----------------</td>
</tr>
<tr>
<td>Multi-node and enterprise deployments</td>
<td>Contact Tableau for technical guidance.</td>
<td>Nodes must meet or exceed the minimum hardware recommendations, except nodes running backgrounder, where 4 cores may be acceptable.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Determining Computer Hardware

To determine how many physical cores a computer has, the Tableau Server setup program queries the operating system. To view hardware information that the setup program detected on your computer, open the `tabadmin.log` file in the following folder on the computer where you are installing Tableau Server:

```
<install directory>\ProgramData\Tableau\Tableau Server-\logs\tabadmin.log
```

In the `tabadmin.log` file, look for lines similar to the following. These lines provide information about the physical and logical cores that the setup program detected and that it used to determine the core count that is being used for licensing.

```
2015-04-09 14:22:29.533 -0700_DEBUG_10.36.2.32:<machine name>_::_pid=21488_0x2cd83560__user=__request=__ Running hardware check

2015-04-09 14:22:29.713 -0700_DEBUG_10.36.2.32:<machine name>_::_pid=21488_0x2cd83560__user=__request=__ Detected 12 cores and 34281857024 bytes of memory

2015-04-09 14:22:29.716 -0700_DEBUG_10.36.2.32:<machine name>_::_pid=21488_0x2cd83560__user=__request=__ Hardware meets recommended specifications. Default values will be used.
```
Manually determining the number of cores on your computer

To determine manually how many physical cores your server has, you can use the Windows Management Instrumentation Command-line tool (WMIC). This is useful if you do not know whether your computer will meet the minimum hardware requirements for installing Tableau Server.

1. Open a command prompt.

2. Enter the following command:

   WMIC CPU Get DeviceID,NumberOfCores

   The output will display the device ID or IDs and the number of physical cores the computer has.

   ![Command Prompt Output]

   In this example, there are two CPUs, each with six cores, for a total of twelve physical cores. This computer would satisfy the minimum hardware requirements for installing Tableau Server.

   The following command shows a longer version that lists the logical processors as well as the physical cores.

   WMIC CPU Get DeviceID,NumberOfCores,NumberOfLogicalProcessors,SocketDesignation
In the above example, the server has a total of twelve physical cores, resulting in 24 logical cores.

Prepare for the Upgrade

To properly prepare for a Tableau Server upgrade, gather information about your existing installation, key files related to your installation and the upgrade, and complete the pre-upgrade tasks.

Gather information and required files

Credentials and permissions

You will need the following credentials in order to upgrade and configure Tableau Server:

- Credentials for a Windows user account that has administrative access on the Tableau Server computer. You need these credentials in order to run the setup program for upgrading and to run `tabadmin`. You also need administrative credentials for worker node computers if your upgrade requires you to upgrade the workers manually.

- Credentials for the Run As User account on the Tableau Server computer.

The Run As User account is granted permissions and access to local server resources. See Required Run As User Account Settings.

In some organizations, group policy or other governing processes may reset permissions granted to the Run As User account. Changes in permissions are likely to cause upgrade
failures. We recommend that you verify the following permissions and security policies before you upgrade:

- Verify Registry Permissions
- Verify Folder Permissions
- Verify the Local Security Policy

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:

- **Customizations** This includes non-default ports, timeout values, custom logo images, and fonts. Also make a note of Windows path environment variables that affect Tableau Server.

- **SMTP configuration.** You can see your current SMTP configuration on the SMTP Setup tab of the Configuration utility. For more information, see Configure SMTP Setup.

- **SSL configuration and certificates.** You can see your SSL configuration on the SSL tab of the Configuration utility. This tab also lists the location of the certificate and certificate key files. You should copy and save these files in a safe location. For more information, see Configure External SSL.

- **SAML configuration, certificates and any IdP metadata files.** You can view your
current SAML configuration on the SAML tab of the Configuration utility, including the certificate, key, and metadata files. Save copies of these files to a safe location. For more information, see Configure SMTP Setup.

- **Kerberos** configuration. You can see your current Kerberos configuration on the Kerberos tab of the Configuration utility, including the location of the keytab file you should copy and save. For more information, see Configure Kerberos.

- **OpenID** configuration. Find your current OpenID configuration details on the OpenID tab of the Configuration utility. For more information, see Configure Tableau Server for OpenID Connect.

- **Worker** configurations. Collect the configurations of your worker nodes, including any certificates or other supporting files that you have needed to copy to the worker nodes. You can find detailed information about the number of processes configured on each node on the Servers tab of the Configuration utility. See Reconfigure Processes for more information.

- **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.

Environment configuration

The steps you need to take during the upgrade process depends 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.
Install location

By default Tableau installs to C:\Program Files\Tableau\Tableau Server. If your current Tableau Server installation is to a non-default path, you need to note the current location in case you are prompted to provide it during upgrade. Depending on your existing version and the version you are upgrading to, the setup program may not find your existing data and configuration unless you specify it.

Setup files

You will need the following setup files before you upgrade Tableau Server:

- The setup program for your existing version of Tableau Server; if you are upgrading a distributed installation, the Tableau Worker Software setup program.

You might not need these. However, we recommend that you have them available in case there’s a problem during the upgrade. That way you can use these and your server backup to restore your installation to its pre-upgrade state.

Note: If you do not have the install program for your existing version, you can download it from the Alternate Downloads Site. If you have a distributed installation of Tableau Server, you should also download the corresponding installer for worker nodes. Save the setup programs in a safe location that is not part of your production or test version of Tableau Server. You will need these if you need to go back to your existing version after upgrading.

- The setup program for the new version of Tableau Server, including the Worker Software Setup program, if you are upgrading a distributed installation.

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.
Pre-Upgrade Tasks

Perform these tasks before starting your upgrade.

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:

- 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.

If you already have a backup that lets you restore the current state of the server, you can skip this step.

1. On the Tableau Server computer, open a command window as an administrator.

2. Change to the \bin folder where the tabadmin.exe file is. For example to change the \bin directory in a default installation:

   ```bash
   cd "C:\Program Files\Tableau\Tableau Server\version\bin"
   ```

3. Type the following command to remove unneeded files from the server:
tabadmin cleanup

4. Type the following command to back up the server, substituting your own path and filename for myserver.tsbak:

   tabadmin backup myserver.tsbak

If you're running Tableau Server version 9.3 or later, run the backup command using the --verify option to verify the backup:

   tabadmin backup --verify myserver.tsbak

For more information, see Remove Unneeded Files and Back Up Tableau Server Data.

- 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. Before upgrading, make sure that the server's maintenance hasn't expired.

1. In Windows, select Start > All Programs > Tableau Server > Manage Product Keys.

2. In the Manage Product Keys dialog box, look for the expiration date under the Maintenance Expires column.

   ![Manage Product Keys](image)

   If your maintenance has expired, select the product key and then click Refresh. If the maintenance date isn't updated, contact Tableau Technical Support. Reactivating the product key will be part of the upgrade process.
For more information, see Activate Tableau Server. If your server doesn’t have internet access, see Activate Tableau Server Offline.

Disable your extracts 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. We recommend that you test the performance of your workbooks that use extracts before you upgrade. For more information, see Test Workbook Performance.

**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

**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.

**Note:** To perform these steps, you must be signed into the Tableau Server computer as a Windows administrator. This applies both to the existing server and to the test server environment.

1. On the existing production environment, create a backup of Tableau Server using the `tabadmin backup` command. If you're using version 9.3 or later, include the --
verify option.

For more information, see Creating 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.

   Note: You can download the setup program for your current version (and the Worker Setup Program for nodes in a cluster) from the Alternate Downloads Site.

3. Restore your existing database without configuration data using the tabadmin restore command and the --no-config option.

   You don't want the existing configuration data because your test server will have different IP addresses, and you don't want the test environment to conflict with the existing server.

   For more information, see Restore from a Backup.

   If you are not the administrator of the production Tableau Server installation and do not have those credentials, you might need to reset the Tableau Server administrator password using the tabadmin reset command.

4. Manually replicate your existing Tableau Server configuration.

   You need to manually configure certain aspects of your environment because you restore the Tableau database using the --no-config option. You also need to manually configure some customizations.

Upgrade the test environment

Follow the appropriate steps for upgrading the test environment, depending on your configuration:
- Upgrade a Single-Node Tableau Server
- Upgrade a Multi-Node (Distributed) Tableau Server

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 worker 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 (tabadmin and tabcmd) and programmatic access via APIs.

**Performance and user acceptance testing**

Use tools like Tabjolt, Replay, 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 Resources. 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.

For more information on new features, see What's New in the Tableau Server Help.

**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 troubleshooting 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 using extracts and have calculations, see Troubleshoot Performance of Workbooks with Calculations.

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. **Setup your test environment**
2. **Capture performance metrics and analyze**
3. **Troubleshoot performance issues**
4. **Capture the conclusions and results**

**Setup your test environment**

1. Create a test environment that mirrors your production environment as closely as possible.
   
   Follow the steps described in the Test the Upgrade topic.

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 tabadmin command line tool to run the following commands:
tabadmin set native_api.node_limit_checker_pre_rewrite_disable false

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, 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 `tabadmin` command line tool to run the following commands:

```
tabadmin set native_api.preserve_calculations_on_hyper_refresh_conversion true
```

```
tabadmin config
```

```
tabadmin restart
```

**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.

5. 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 enable the `-tabadmin set native_api.preserve_calculations_on_hyper_refresh_conversion true` setting, remember to disable your extract refreshes just before you upgrade your production Tableau Server.

After upgrade, enable the `-tabadmin set native_api.preserve_calculations_on_hyper_refresh_conversion true` setting, and re-enable your extract refreshes once you have confirmed that you are not experiencing slow response times after upgrade.

### Perform the Upgrade

After you’ve completed the Prepare for the Upgrade, upgrade your existing Tableau Server installation to version 10.5 by following one of the topics listed at the bottom of this page. (If you are migrating to new hardware as part of your upgrade, refer to Migrate to New Hardware instead.)

When you install the newer version of Tableau Server, use the same drive and directory that the earlier version used. This way, data and configuration settings from your earlier version can be automatically imported. If your installation location is not the default, see Upgrade Tableau Server to a Non-Default Location.

Before upgrading, read through the list of pre-upgrade tasks to make sure you have followed the recommended best practices. For details, see Prepare for the Upgrade.
If you are upgrading from 32-bit Tableau Server to 64-bit Tableau Server you must uninstall your existing version before installing the new version. For more information, see Upgrade from 32-bit to 64-bit Tableau Server.

As a best practice, you should always make a backup of your Tableau Server data before upgrading. This backup is necessary in the event that something unexpected happens during the upgrade, or if you need to roll back to your previous version of Tableau Server. If you create this backup yourself, you can do it while Tableau Server is running. If you are upgrading to version 10.0 or later you can choose to have the Setup program create the backup while upgrading Tableau Server, but the backup is created while the server is stopped so this extends the length of time the server is unavailable. For more information, see Prepare for the Upgrade, Tableau Server Upgrade Backup Options, and Back Up Tableau Server Data.

Upgrade a Single-Node Tableau Server

The instructions in this topic explain how to perform a Tableau Server upgrade on a single-node server, without changing the hardware or other configuration as part of the upgrade. If this is not your situation, see the following topics for instructions:

- If you are upgrading a distributed (multi-node) Tableau Server installation, see Upgrade a Multi-Node (Distributed) Tableau Server.

- If you are upgrading to a non-default location, see Upgrade Tableau Server to a Non-Default Location.

- If you are upgrading to new hardware, see Migrate to New Hardware.

- If you are upgrading from a 32-bit version of Tableau Server to a 64-bit version, see Upgrade from 32-bit to 64-bit Tableau Server.

Before you upgrade

Make sure you’ve prepared for the upgrade by reviewing changes in the new version and by gathering required information. For more information, see Research the Upgrade and Prepare for the Upgrade. Also make sure you’ve downloaded the Tableau Server setup...
program for the primary server node from the download site. Optionally (but recommended),
test the upgrade on a test environment that mimics your production environment. For more
information, see Test the Upgrade.

Perform the upgrade

1. Run the Tableau Server setup program on the Tableau Server computer.
   The setup program will determine that you have a previous version installed and
   prompt you for a backup option.

2. Specify whether you want the setup program to perform a full backup before it unin-
   stall the existing version. This backup is only needed if something unexpected hap-
   pens during the upgrade. The server will be unavailable while the backup is being
   created. If you have a recent backup you can choose the option to not do the full
   backup. For more information, see Tableau Server Upgrade Backup Options.

3. After uninstalling the existing version (and creating a backup, if you selected that
   option), the installation process prompts you for an install location. If you are upgrad-
   ing a version that was installed to the default location, you can accept the default. If
   you are upgrading an installation that was installed to a non-default location, navigate
   to the original location.

   If you do not navigate to the original location, your data and configuration settings from
   the original installation will not be found and used for your upgraded installation.

   **Note:** If you are upgrading from a version of Tableau Server that is earlier than
   9.x, your existing extracts will be migrated to the new File Store during upgrade.
   This process may take a long time (up to several hours if you have a large number
   of extracts or extracts that are large in size). While this takes place a message
displays: "**Migrating extracts to File Store This process may take up to several hours.**" For more information, see Troubleshoot Tableau Server Install and
   Upgrade
Upgrade a Multi-Node (Distributed) Tableau Server

In a multi-node (distributed) installation of Tableau Server, you need to upgrade all the nodes (primary and all workers) so they are running the same version of Tableau.

Start by upgrading the primary node. For some upgrade paths (for example, from version 9.3 to version 10.0), the upgrade process will automatically push an upgrade from the primary node to the worker nodes. If the workers cannot be automatically upgraded, usually when the upgrade includes updates to PostgreSQL drivers or other third-party software, the Setup program will let you know that you must manually upgrade the worker nodes.

**Important:** You should not upgrade a worker node before upgrading the primary node. Start the upgrade on the primary node, and if prompted by the setup process, move to the worker nodes and manually upgrade them before returning to the primary node to complete the upgrade. If you upgrade a worker node before upgrading the primary node, you will need to completely uninstall Tableau Server, reinstall, and restore from your backup.

- If you are upgrading to a non-default location, see Upgrade Tableau Server to a Non-Default Location.

- If you are upgrading to new hardware, see Migrate to New Hardware.

- If you are upgrading from a 32-bit version of Tableau Server to a 64-bit version, see Upgrade from 32-bit to 64-bit Tableau Server.

Upgrading a high availability installation of Tableau Server requires some additional steps to upgrade the backup primary, as explained later in this topic.

Before you upgrade

Make sure you’ve prepared for the upgrade by reviewing changes in the new version and by gathering required information. For more information, see Research the Upgrade and Prepare for the Upgrade. Also make sure you’ve downloaded the Tableau Server setup
program and Worker Software setup program from the download site. Be sure to download the same version for both primary and workers. You will need the worker software if it turns out you must manually upgrade worker nodes. Optionally (but recommended), test the upgrade on a test environment that mimics your production environment. For more information, see Test the Upgrade.

Generally there are no changes to the server upgrade experience for the new Data Engine. If you are upgrading a multi-node server, you will be prompted to manually upgrade the additional nodes. Upgrade may affect your current configuration for a multi-node cluster, especially if you have certain processes isolated on a separate node. For example, if you upgrade a node that has just the Backgrounder installed, after upgrade you will have one instance of Data Engine installed on that node as well. For more information about the new Data Engine and the configuration changes, see The New Data Engine in Tableau Server.

Upgrade the primary node and worker nodes

1. Run the Tableau Server setup program on the primary server node.

   The setup process will determine that you have a previous version installed and prompt you for a backup option.

2. Specify whether you want the setup program perform a full backup before it uninstalls the existing version. This backup is only needed if something unexpected happens during the upgrade. The server will be unavailable while the backup is being created. If you have a recent backup you can choose the option to not do the full backup. For more information, see Tableau Server Upgrade Backup Options.

3. After uninstalling the existing version (and creating a backup, if you selected that option), the installation process prompts you for an install location. If you are upgrading a version that was installed to the default location, you can accept the default. If you are upgrading an installation that was installed to a non-default location, navigate to the original location.
If you do not navigate to the original location, your data and configuration settings from the original installation will not be found and used for your upgraded installation.

4. With most upgrades, the worker nodes will be automatically updated.

**Note:** When worker nodes are upgraded automatically, the Windows registry on the worker nodes will not reflect the upgrade, so the old version will still show in the program list of Control Panel. This is expected.

If the upgrade process requires you to upgrade the worker nodes manually, you will be prompted during installation of the primary with a message: "One or more workers could not be upgraded automatically."

a. If you see this message, sign in to each worker node and run the Tableau Worker setup program.

b. Setup uninstalls the existing version of the worker node software and installs the new version.

c. Return to the primary server.

5. Follow the prompts in the Setup program to complete the installation.

**Note:** If you are upgrading from a version of Tableau Server earlier than 9.x, your existing extracts will be migrated to the new File Store during upgrade. This process may take a long time (up to several hours if you have a large number of extracts or extracts that are large in size). While this takes place a message displays: "**Migrating extracts to File Store This process may take up to several hours.**" For more information, see Troubleshoot Tableau Server Install and Upgrade

If you are upgrading a distributed installation that includes a backup primary, you need to upgrade the backup primary in a separate step, after upgrading the rest of the cluster.
Upgrade a backup primary server

If you have configured your Tableau Server installation for high availability, after you've finished the upgrade for the primary node and the worker nodes, upgrade the backup primary computer.

1. On the backup primary, run the Tableau Server Setup program.

   Setup will determine that you have a previous version installed and prompt you for a backup option. Choose to not create a backup.

2. If you changed hardware or any configuration values on the primary or worker nodes during the upgrade, reconfigure the backup primary by following the instructions in Create a Backup Primary.

Tableau Server Upgrade Backup Options

When you upgrade Tableau Server, the setup program can make a backup as part of the upgrade process. This can be helpful in case something unexpected happens while you are upgrading, or if you need to go back to your previous version.

**Note:** Beginning with version 10.0 of Tableau Server, the Setup program gives you the option to skip making a full backup.

You might choose instead to create a backup before you start the upgrade process and then skip the backup during the upgrade. This lets you create a backup while Tableau Server is running and available to your users. If the backup occurs during the setup process, the server is stopped, and it increases the length of time your users cannot access their Tableau content.

For more information on backing up Tableau Server, see Back Up Tableau Server Data. For more information about restoring a backup, see Restore from a Backup.

If the setup process detects an existing installation of Tableau Server, it offers you the following options:
• **Full backup.** When you select this option, the setup process performs a complete backup of your Tableau data and configuration before it uninstalls the existing version.

Choose this option if you haven't recently made a backup and it is acceptable for Tableau Server to be unavailable to your users during the backup. A full backup can take a significant amount of time—up to several hours if you have a large installation or a lot of stored data (extracts). Because the server is unavailable during the period the backup is being created, choose this option only if you do not already have an up-to-date backup.

• **Without full backup.** When you select this option, the upgrade process uninstalls the previous version without making a backup.

Choose this option only if you have a recent backup. This option can save you a significant amount of time during the upgrade (for data-heavy installations, you can save hours).

**Note:** 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.

### Special Installation Scenarios

The bulk of the upgrade instructions cover typical upgrade situations. Not everyone is doing a straightforward or default upgrade, and the following topics cover some of the most common non-standard situations.

#### Upgrade from 32-bit to 64-bit Tableau Server

Starting with version 10.0, Tableau Server is available only as a 64-bit application. In previous versions, it was also available as a 32-bit application.
A distributed installation of Tableau Server must run the same bit version (all 32-bit or all 64-bit) and release version (10.0 for example) on all nodes. When you upgrade a distributed installation of Tableau Server to the 64-bit version, you need to manually upgrade each worker node by uninstalling the 32-bit version on each worker before you install the 64-bit version of the worker software.

1. Uninstall the 32-bit version on your primary Tableau Server computer.

Note: Uninstalling removes the server software but leaves your data and configuration settings intact. If your existing 32-bit version was installed to the default location (C:\Program Files (x86)\Tableau\Tableau Server-<version>) the 64-bit Setup program will find the data and configuration and use it in the upgraded installation. If your existing 32-bit version was installed to a non-default location, see Upgrade Tableau Server to a Non-Default Location.

2. Install 64-bit Tableau Server on the primary Tableau Server node.

3. Uninstall the 32-bit version on each worker node.

4. Install 64-bit Tableau Server Worker software on each worker node.

5. Return to the primary server and complete the configuration of 64-bit Tableau Server.

Upgrade Tableau Server to a Non-Default Location

You need to install Tableau Server to the same location as your existing Tableau Server so the setup process can locate and upgrade the existing configuration and data. By default, the Tableau Server setup program installs Tableau Server in the following location:

C:\Program Files\Tableau\Tableau Server\version

and stores data and logs in this location:

C:\ProgramData\Tableau\Tableau Server
If your existing installation is in the default location, you can accept the defaults in the Setup program, and the install process finds your associated data and configuration settings during upgrade.

If your existing installation is not in the default location, you need to tell the Setup program where to install the new version of Tableau Server. How you specify a non-default upgrade location when prompted by the Setup program depends on whether you browse to the location or type the path.

- **If you browse to the install location**, the Setup program takes that path and appends \Tableau Server to it, and then installs to a `<version>` folder immediately below the \Tableau Server folder. You can see the path that Tableau Server will be installed to when you are browsing, and you can edit that path if it is not correct (if, for example, you do not have a \Tableau Server folder).

- **If you type the install location**, the Setup program accepts the path you type and installs to a `<version>` folder at the end of that path.

Don't include the `<version>` folder when browsing or typing the path.

For example, to upgrade an existing 10.0 Tableau Server installed to D:\Tableau\10.0, you can either

- Browse to D: (the Setup program will populate the path as D:\Tableau Server) and edit the path so it reads D:\Tableau

  or

- Type the path as D:\Tableau

**Migrate to New Hardware**

Use the following procedure to migrate Tableau Server from one computer to another. Specifically, these steps describe how to move Tableau Server data and configuration settings 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. The backup is what you use to restore your Tableau Server data.

1. Install Tableau Server on the new computer. For details, see Run Server Setup.

2. Copy your backup file .tsbak to the bin folder on your new Tableau Server (for example, C:\Program Files\Tableau\Tableau Server-\<version>\bin).

   **Note:** The file does not need to be in the bin directory but if it is not, you will need to include a full path to the file when you run the restore command.

3. Next, stop Tableau Server.

4. Restore your in-production data without configuration information to your new Tableau Server installation using the tabadmin restore command:

   ```
tabadmin restore --no-config <filename>
   ```

   where `<filename>` is the name of the .tsbak file. For example:

   ```
tabadmin restore --no-config mybackup.tsbak
   ```

   The --no-config option restores the data from your in-production Tableau Server but excludes configuration information. You need to use this option when moving to new hardware to avoid conflicts with the old configuration. After doing the restore, you may need to reconfigure some options (SMTP or proxy settings, for example).

   For more information about restoring Tableau Server data, see Restore from a Backup.

5. Start the server.

6. **Distributed installations only:** Run the Tableau worker installer on the new worker computers you want to add to your new Tableau Server cluster. See Install and
Configure Worker Nodes for steps.

7. 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:

a. Select **Start > All Programs > Tableau Server > Manage Product Keys**.

b. For each product key, select the product key and click **Deactivate**.

*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 for steps.

---

**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 indexing and processes not fully started can be resolved by restarting Tableau Server in a controlled way. To restart Tableau
Server, use the `tabadmin restart` command. This will stop all the processes associated with Tableau Server and then restart them.

3. Clean up files associated with the Coordination Service (ZooKeeper). To clean up Coordination Service files, use the `tabadmin cleanup --reset-coordination` command.

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 `tabadmin 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 `tabadmin restart` command from a command prompt. This will shut down any processes that are running, and restart Tableau Server.

Installing Tableau Server

Install fails due to hardware requirements

Starting with version 9.0, Tableau Server cannot install if the computer you are installing on does not meet the minimum hardware requirements. The requirements apply to both primary server computers and worker computers. For details on minimum hardware requirements, see Minimum Hardware Requirements and Recommendations for Tableau Server.
Install or upgrade generates an error when PostgreSQL ODBC driver does not install correctly.

In certain circumstances (when a system reboot is pending, or another program is being installed or updated, the Tableau Server PostgreSQL ODBC driver does not install correctly. When this happens, this message displays:

PostgreSQL ODBC driver (64-bit) version 09.03.0400 did not install properly.

Note: The version may be different, depending on what version of Tableau Server you are installing.

If this occurs, follow these steps to correct the issue:

1. Check to see if the driver shows as installed in Control Panel.
2. If the driver is not installed, download it from the Tableau Drivers page and install it.
3. If the driver is installed, uninstall it from Control Panel, restart the computer, download the driver, and install it again.

Upgrading Tableau Server

Extract migration is slow

Tableau Server 9.0 introduced a more reliable storage mechanism for data extracts called the File Store. Upgrading from a previous version requires migration of the extracts. This can take a long time (up to several hours) if you have a large number of extracts or extracts that have a lot of data. During migration a message displays:

Migrating extracts to File Store
This process may take up to several hours.

If the migration progress appears to be stalled or stuck, you can verify that migration is continuing by watching the tabadmin.log. An entry is written to this log for each extract that
is migrated. You can periodically copy the log and open your copy in a text editor like Note-pad to verify that entries are being written to it.

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. Zip and save logs using the `tabadmin 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 `tabadmin cleanup` command. For more information, see Remove Unneeded Files

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 `tabadmin reindex` command.
Activating Tableau Server

Tableau Server license activation fails

In some instances Tableau Server license activation may fail with one of the following error 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.tableausoftware.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
- o.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
Requests to the above domains may be on port 80 or 443.

Verify the date and time

Verify the date and time on the primary 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.

Verify FlexNet Licensing Service has started

If the date and time on the Tableau Server computer are correct, verify that the FlexNet Licensing Service is running on the primary Tableau Server computer.

1. On the primary computer, from the Windows Start menu, open services.msc.

2. In the Services dialog box, verify that the status of FlexNet Licensing Service 64 (64-bit) or FlexNet Licensing Service (32-bit) is Started.

   If FlexNet is not listed as Started, right-click FlexNet Licensing Service and select Start.
If the Start option is grayed out, the service may be set to Disabled. To enable the service:

a. Right-click FlexNet Licensing Service and select Properties.

b. From the Startup type drop-down list, select Automatic.

c. Click the Start button, and then click OK.

Force the product key to be read again

1. On the primary Tableau Server computer, sign in as administrator and open a command prompt.

2. Change to the Tableau Server bin directory. By default this is:
   
   C:\Program Files\Tableau\Tableau Server\10.5\bin

3. Type the following commands:
   
   tabadmin stop
   lmreread
   tabadmin 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 primary Tableau Server computer, sign in as administrator and open a command prompt.

2. Change to the Tableau Server bin directory. By default this is:
   
   C:\Program Files\Tableau\Tableau Server\10.5\bin

3. Type the following command:
serveractutil -view > LicResults.txt

4. Contact Customer Support [http://www.tableau.com/support/request] and include the LicResults.txt file that you created.

Navigate Server Admin Pages

As a server administrator, you can access all of the menus and pages in Tableau Server for server and site management. If your server is configured for multiple sites, the site menu is available for navigation. Click Manage All Sites in the site menu to access server administration pages.

The server administrator pages include server-wide settings that you will use to configure, monitor, and maintain Tableau Server.

Server Administrator Pages

In a single-site deployment, all server and site menus are available to you in the main menu. To create a site, click Settings > Add a Site.

On a multi-site server, when a site is selected, you will see these menus:

To access server administration pages, click the site menu, and then select Manage All Sites:
On a multi-site server, these are the server administration menus. The site menu text changes to **All Sites** to let you know you are managing server-wide settings.

To return to the site administration menus, click **All Sites**, and the select the site you want to manage.
Server administrators can:

- 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 site-related settings, you must first navigate to a specific site. Within each site, you can:

- Manage content and assign permissions.
- 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).

**Move Tableau Server to Another Drive**

By default Tableau Server is installed to the system drive (usually C:\). This means that all program and data files are installed to that drive. If you need to move Tableau Server to a different drive (if you are running out of space on the original drive, for example), you can do this by following the procedure below. The steps are intended as an example for moving Tableau Server from one drive to another drive on the same computer, and may not exactly reflect your installation and configuration.

**Before you start**

Before starting, make sure you:

• Have administrative credentials to the primary Tableau Server computer. This is where you will be working and you need to be logged on to the computer as an administrator.

• Have a copy of the installation program for the version of Tableau Server you are moving. You need this to install Tableau Server to the new drive.

• Know what authentication methods your current installation is using. If Tableau Server is configured for SSL, SAML, or Kerberos, you will need to back up the related certificate or keytab files separately, and then copy them to the new drive after you reininstall Tableau Server.

When you are ready to move Tableau Server to another drive:
1. Open a command prompt as an administrator on the primary Tableau Server computer.

2. Navigate to the Tableau Server \bin directory. By default this is:
   
   \C:\Program Files\Tableau\Tableau Server\<version>\bin

3. Stop Tableau Server:
   
   tabadmin stop

4. Back up your Tableau Server data:
   
   tabadmin backup <filename>

   Where <filename> is the name of the backup file. The file will have a .tsbak extension.

   Save the backup file to a safe location. You need this to restore your server after installing it to the new drive.

5. If Tableau Server is configured for SSL, SAML, or Kerberos, make a copy of the folder that contains the related certificate or keytab files and save it to a location that is not part of the Tableau Server install path (for example, C:\TServer-hold). Find the location of the related files by opening the Tableau Server Configuration utility and clicking the appropriate tab.

6. Uninstall Tableau Server u using Control Panel and Add/Remove programs. You should also rename the \ProgramData\Tableau\Tableau Server and \Program Files\Tableau\Tableau Server\<version> directories so your new installation does not find them. By default the \ProgramData directory is hidden, so you may need to configure Windows to display hidden files and directories to find it. Later, once you have installed Tableau Server to the new location and verify that everything is working, you can delete the directories.

7. Install Tableau Server to the new drive.
At the end of the install process a browser window will open and you will be prompted to create an initial administrative user. Close this without creating the user.

8. Open a command prompt as an administrator.

9. Navigate to the Tableau Server \bin directory in your new install location.

10. Restore your Tableau Server data backup from Step 4 above using the --no-config option:

    tabadmin restore --no-config <filename>

    Where <filename> is the name of the .tsbak file you created when you backed up your original installation.

11. If you had SSL, SAML, or Kerberos configured, copy the folder with your certificates to the new drive. For example:

    a. If you are using SSL and installed Tableau Server to D:\Tableau\Tableau Server, copy the SSL folder to D:\Tableau\Tableau Server\SSL.

    b. Run the Tableau Server Configuration utility and on the SSL tab, update the paths to the new location for the certificate files.

    c. Click OK to close the Configuration utility.

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 license keys. However, you can manually activate the license 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 Windows

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 is not possible you’ll need to download the required packages to a trusted computer outside the air gap and transfer them to the air-gapped computer:

1. On a trusted computer outside the air gap with internet access, download the Tableau Server installation package.

2. Transfer the package to your removable media.

3. On your air-gapped computer, insert the removable media containing the Tableau Server installation package, and then run the installer.

4. After installation is complete, you can activate the Tableau Server license 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.

1. On your Tableau Server in the air-gapped environment, when the product key manager opens, click **Activate the product**.

2. Enter your server product key into the corresponding text box and click **Activate**. You can get your product key from the Tableau Customer Portal.

3. When you are offline, activation will fail. Tableau Server provides the option to save a file that you can use for offline activation. Click **Save**.

4. Select a location for the file and click **Save**. The file is saved as **offline.tlq**.

5. Click **Exit** to close the **Activation** dialog box.

6. Copy the offline activation file (**offline.tlq**) from the target directory to a trusted computer that has internet access.

7. 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 the **server_firstpass_windows.tlq** file. The file is a simple XML file.

8. Open both **offline.tlq** and **server_firstpass_windows.tlq** in a text editor and turn on line numbering.

9. Update the following XML elements in **server_firstpass_windows.tlq** with the values for the same elements in **offline.tlq** on the disconnected computer. 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 - <MachineIdentifier> (Optional: If this XML element is present, copy the value. If it's missing you can skip it.)

Line 11 - <Value>

Line 12 - <Value>

Line 14 - <SequenceNumber>

Line 60 - <Hash>

10. Upload the edited template (offline.tlq) to the Tableau Offline Activation website.

    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_windows.tlq file. It will match all the places to put the values from the generated .tlq file.

11. Update the following XML elements in server_secondpass_windows.tlq with the values for the same elements in offline.tlq on the disconnected computer. 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>
12. Upload the edited template (offline.tlq) to the Tableau Offline Activation website.

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 AWS 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 Tableau Server in the Amazon Web Services (AWS) cloud using Amazon Elastic Compute Cloud (Amazon EC2) instances. Running Tableau Server on AWS is an excellent choice if you want the flexibility of scaling up and scaling out without having to purchase and maintain an expensive fleet of servers. Tableau Server can take advantage of many native AWS services. For example, you can use the following services in AWS with Tableau Server:

- **Amazon CloudWatch**: Monitors the components of your AWS-based solution.
- **Amazon Route53**: DNS web service for connecting user requests to your AWS infrastructure.
- **AWS Certificate Manager**: Helps you deploy SSL and TLS certificates for use with AWS.
- **AWS CloudFormation**: Helps you to create and manage related AWS resources in a
controlled manner using pre-defined templates.

- **Elastic Load Balancing**: You can use Elastic Load Balancing to spread Tableau requests across multiple Tableau gateways.

In addition, you can store the data you use with Tableau Server using any of the following AWS features. Tableau provides native data connectors that enable you to connect to data in the following AWS data sources:

- **Amazon Athena**: Interactive query service that you can use to analyze data in Amazon S3 using SQL.
- **Amazon Aurora**: High-performance MySQL- or PostgreSQL-compatible database.
- **Amazon EMR**: Managed Hadoop framework for processing large amounts of data.
- **Amazon Redshift**: Data warehousing solution for storing and mining large amounts of data.
- **Amazon Relational Database Service (Amazon RDS)**: Relational database solution supports Amazon Aurora, PostgreSQL, MySQL, MariaDB, Oracle, and Microsoft SQL Server.
- **Amazon Simple Storage Service (Amazon S3)**: Cloud-based storage solution.

### Tableau Server on AWS deployment options

You can self-deploy Tableau Server on an Amazon EC2 instance that you provision, deploy Tableau Server using the AWS CloudFormation templates in the Tableau Server on AWS Quick Start, or deploy Tableau Server on AWS using an AWS Marketplace Amazon Machine Image (AMI). The following table shows the differences between each option.

<table>
<thead>
<tr>
<th>Tableau Server</th>
<th>AWS Self-Deployement</th>
<th>AWS Quick Start</th>
<th>AWS Marketplace AMI</th>
</tr>
</thead>
<tbody>
<tr>
<td>Production ready</td>
<td>✅</td>
<td>✅</td>
<td>✗</td>
</tr>
<tr>
<td>Upgradable</td>
<td>Install on Windows</td>
<td>Install on Linux</td>
<td>Scale-up</td>
</tr>
<tr>
<td>------------</td>
<td>-------------------</td>
<td>------------------</td>
<td>----------</td>
</tr>
<tr>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
<td>✔️</td>
</tr>
<tr>
<td><strong>Each of the deployment options are described in more detail below:</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- **Self-Deploy to an Amazon EC2 Instance** – Provides the most flexibility and options to customize Tableau Server to your environment. If you want to quickly launch a system use the Tableau Server on AWS Quick Start or an AWS Marketplace AMI instead.

  Self-deployment of Tableau Server to an Amazon EC2 instance is recommended for development, test, and production environments that start simple but may need to scale-up or scale-out later.

  For more information about self-deployment, see [Self-Deploy a Single Tableau Server on AWS](#). For more information about deploying in a distributed environment, see [Self-Deploy Tableau Server on AWS in a Distributed Environment](#).

- **Tableau Server on AWS Quick Start** – Installs Tableau Server on a standalone
Amazon EC2 instance or on a cluster of standalone Amazon EC2 instances using an AWS CloudFormation template. This deployment strategy allows you to go live quickly and limit your overall costs.

The Tableau Server on AWS Quick Start is recommended for development, test, and production environments that start simple but may need to scale-up or scale-out later. You can use your own license (BYOL) or a free 14-day trial version.

For more information about installing Tableau Server on Windows using the Quick Start, see the Tableau Server on AWS Quick Start.

- **Tableau Server AWS Marketplace AMIs**—Installs Tableau Server on Windows with an AWS Marketplace AMI. You can use a 14-day free trial or after Tableau Server is up-and-running on the Amazon EC2 instance, you can apply the Tableau Core or Interactor license you’ve purchased to the server. The Tableau Server AWS Marketplace AMIs are recommended for quick testing and proof-of-concept environments only. For more information about AWS Marketplace AMIs, see the AWS Marketplace. The Tableau Server AWS Marketplace AMIs have the following limitations:
  
  - Choose BYOL, 10, 25, 50 or 100 user AMIs running Microsoft Windows.
  
  - Tableau Server license included in the hourly fee paid to AWS. If you’re using the Bring Your Own License (BYOL) AMI, the license is sold separately.
  
  - Single machine, scale-up (vertical scaling) only.
  
  - No Active Directory. Local authentication only. If you need Active Directory authentication, see Self-Deploy to an Amazon EC2 Instance.
  
  - No in-place upgrades. You may not upgrade this software to a newer version of Tableau Server. Instead, you must backup Tableau Server and restore your users and content to a newer AMI running Tableau Server.
AMIs may be deployed to a specific set of optimal Amazon EC2 instance types and sizes.

To install an AWS Marketplace AMI

2. In the search box enter Tableau.
3. Select the Tableau Server product you want to use. When prompted, sign in to your AWS account.
4. Review the product details, and then click Continue when you are ready to deploy Tableau Server on AWS.
5. On the AMI details page, click the 1-Click Launch tab.
   You can configure all the settings for your instance on this tab. This option allows you to launch only one Amazon EC2 instance at a time.

   **Note:** If you prefer to use the Amazon EC2 console to launch your instance, click the Manual Launch tab from the AMI details page. This option allows you to launch multiple Amazon EC2 instances.

6. In the Region section, select the AWS region where you want to launch your Tableau Server instance.
7. In the EC2 Instance Type section, choose your preferred instance type. See the previous section for recommendations.
8. In the VPC Settings section, choose the Amazon Virtual Private Cloud (Amazon VPC) that your instance will be launched in.
To ensure that your VPC is secure, see Security in the Amazon Virtual Private Cloud User Guide at the AWS website.

9. In the Security Group section, review the proposed security group settings and either choose from your existing security groups or accept the new proposed security group.

10. In the Key Pair section, choose a key pair to associate with the instance. This key is used to connect to your instance over Secure Shell (SSH) or to decrypt your Windows administrator password.

11. Click Accept Terms & Launch with 1-Click to create your new instance.

What You Need Before You Begin

Before deploying Tableau Server on AWS, you must have the following:

- An AWS account.

- An Amazon EC2 key pair.

For more information, see Amazon EC2 Key Pairs and Windows Instances in the Amazon EC2 User Guide for Windows Instances at the AWS website.

- A Tableau Server trial license or product key.

  - For a standalone deployment, you can use a 14-day trial version of Tableau Server, which doesn’t require the use of a product key. After 14 days, you’ll need to purchase a license, or use a subscription license.

  - For a cluster-based deployment, you need a user-based server license (which covers all authorized users of Tableau Server), a core-based server license (with a minimum of 16 cores), or a subscription license. To obtain a product key, contact sales.

- (Optional) A domain managed by Amazon Route 53.
• (Optional) An SSL certificate managed by AWS Certificate Manager in the region where you are deploying Tableau Server.

• Storage on Amazon EC2 instance (a general purpose SSD (gp2) is recommended)

• (Optional) An Elastic IP address if you aren’t using ELB or Amazon Route53.

Best Practices for Installing Tableau Server on AWS

The following best practices make installing Tableau Server in the cloud a better experience.

• Read the Tableau Server Plan Your Deployment section.

• Read What You Need Before You Begin.

• Read the Tableau Server Security Hardening Checklist.

• If you’re new to the cloud, use the AWS Free Tier to get familiar with working in the cloud before you deploy Tableau Server on AWS.

• Read about and understand network topology on AWS and how to architect it.

Keeping Costs Down

AWS offers cloud-based services on a pay-as-you-go basis. Costs are determined by the services you run and amount of time you use them. Different combinations of instance types and sizes have different costs. For more information services pricing, see AWS Pricing at the AWS website. You can estimate your total monthly costs using the AWS Simple Monthly Calculator at the AWS website. You can also compare on-premises vs the cloud using the AWS Total Cost of Ownership (TCO) Calculators at the AWS website.

To help monitor and control usage costs on an ongoing basis, you can use Amazon CloudWatch to alert you when your monthly AWS costs reach your predefined spending threshold. For more information, see Monitor Your Estimated Charges Using Cloudwatch in the Amazon CloudWatch User’s Guide at the AWS website.
Tableau Server on AWS Topology

When deploying Tableau Server on AWS, you may choose the level and depth of integration between Tableau Desktop and Tableau Server on your Amazon EC2 instance(s). You can use Tableau Desktop and its ability to extract data from your data sources to act as a bridge between your data and Tableau Server. Based on your needs, or if you already have a lot of data in the AWS cloud, you might instead choose to leverage the full range of AWS services in conjunction with Tableau Server. The following diagram shows data source integration with Tableau Desktop and Tableau Server.

In the following diagram, all of your data is hosted on AWS. You can analyze both structured and unstructured data managed in a secure, scalable manner. You may leverage data that resides on AWS only, unmanaged data that lives outside of AWS, or a combination of both. This flexibility greatly increases the ability of your organization to support moves to the cloud from on-premise data, since both types of data is supported equally.
Selecting an AWS Instance Type and Size

Choosing the right instance for your workload is an important factor for a successful Tableau Server deployment. You can choose from a wide range of Amazon EC2 instance types. For a complete list of all available instance types and sizes, see Amazon EC2 Instance Types at the AWS website.

At minimum, a 64-bit Tableau Server requires a 4-core CPU (the equivalent of 8 AWS vCPUs) and 8 GB RAM. However, a total of 8 CPU cores (16 AWS vCPUs) and 64GB RAM are strongly recommended for a single production Amazon EC2 instance.

An AWS vCPU is a single hyperthread of a two-thread Intel Xeon core for M4, M3, C4, C3, and R3 instances. A simple way to think about this is that an AWS vCPU is equal to half a physical core. Therefore, when choosing an Amazon EC2 instance size, you should double number of cores you have purchased or wish to deploy with. Example: You have purchased an 8 core license for Tableau Server (or need to support enough active users where 8 cores are warranted). You should choose an Amazon EC2 instance type with 16 vCPUs.

The Windows Operating System will recognize these 16 vCPU as 8 cores, so there is no negative licensing impact.
Typical instance types and sizes for development, test, and production environments

- m4.4xlarge
- r4.4xlarge
- r4.8xlarge

Recommended specifications for a single production instance

<table>
<thead>
<tr>
<th>Component/Resource</th>
<th>Amazon AWS</th>
</tr>
</thead>
<tbody>
<tr>
<td>CPU</td>
<td>16+ vCPU</td>
</tr>
<tr>
<td>Operating System</td>
<td>Windows:</td>
</tr>
<tr>
<td></td>
<td>• Windows Server 2012r2, 64-bit</td>
</tr>
<tr>
<td></td>
<td>• Windows Server 2016, 64-bit</td>
</tr>
<tr>
<td>Memory</td>
<td>64+ GB RAM (4GB RAM per vCPU)</td>
</tr>
<tr>
<td>Storage</td>
<td>Two volumes:</td>
</tr>
<tr>
<td></td>
<td>30-50 GiB volume for the operating system</td>
</tr>
<tr>
<td></td>
<td>100 GiB or larger volume for Tableau Server</td>
</tr>
<tr>
<td>Storage type</td>
<td>EBS recommended (SSD (gp2) or Provisioned IOPS)</td>
</tr>
</tbody>
</table>

Self-Deploy a Single Tableau Server on AWS

For the most flexibility in security, scaling, and capacity, you can perform a self-deployment of Tableau Server on AWS. 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.

When you deploy Tableau Server in a cluster in AWS, we recommend that you use Amazon Elastic Block Store (EBS) volumes with Amazon EC2. EBS provides persistent block-level storage volumes for use with Amazon EC2 instances. You should attach a separate EBS General Purpose (SSD) volume with a volume size of at least 100 GiB to allow room for installing Tableau Server. This volume size supports a baseline performance of 300 input/output operations per second (IOPS) with the ability to burst to 3,000 IOPS. For more information, see General Purpose SSD (gp2) Volumes in the Amazon EC2 User Guide for Windows Instances at the AWS website.

For heavy workloads where the storage subsystem must provide an absolute minimum number of IOPS for performance, you can either create a larger EBS General Purpose (SSD) volume to get up to 10,000 IOPS with the ratio of 3 IOPS per GiB, or use an EBS Provisioned IOPS (SSD) volume with the IOPS you need. In either case, we recommend that you use EBS-optimized instances if you plan to run heavy extracts and do a lot of data processing. For more information about Tableau Server performance, see Tableau Server Performance Overview.

Perform the following tasks to install and configure Tableau Server in the AWS cloud.

<table>
<thead>
<tr>
<th>Step 1: Create a Virtual Private Cloud (VPC)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Step 2: Configure networking and security</td>
</tr>
<tr>
<td>Step 3: Launch an Amazon EC2 instance</td>
</tr>
<tr>
<td>Step 4: Create an elastic IP address for the VPC</td>
</tr>
<tr>
<td>Step 5: Log in to Amazon EC2</td>
</tr>
<tr>
<td>Step 6: Install Tableau Server</td>
</tr>
</tbody>
</table>

**Step 1: Create a Virtual Private Cloud (VPC)**
Your first step is to create a Virtual Private Cloud (VPC).

1. Sign in to AWS and then navigate to the Amazon VPC console (https://console.aws.amazon.com/vpc/)

2. Use the region selector to choose the location where your other AWS resources are located. Often, this is the region closest to you.

3. Under Resources, click Start VPC Wizard, on the Step 1: Select a VPC Configuration screen, click VPC with a Single Public Subnet, and then click Select.

4. On the Step 2: VPC with a Single Public Subnet screen, give your VPC a name, and then click Create VPC. You can leave all other settings at their default values.

Step 2: Configure networking and security

To allow inbound traffic to reach your VPC, you should limit traffic to three standard ports (HTTP, HTTPS, and RDP). For more information, see Recommended Network ACL Rules for your VPC in the Amazon Virtual Private Cloud User Guide at the AWS website.

1. Navigate to the Amazon EC2 console (https://console.aws.amazon.com/ec2/).

2. Use the region selector to choose the location where you created your VPC.

3. In the navigation pane, click Security Groups, and then click Create Security Group.

4. On the Create Security Group screen, in the Security group name field, enter a name for your security group.

5. In the Description field, enter a description for the security group.

6. In the VPC field, select your VPC from the list.

7. Click the Inbound tab, click Add Rule, and then in the Type list, select HTTP (80),
and in the **Source** column, choose **My IP** for each rule. This limits inbound traffic to your computer. To specify an IP address range instead, choose **Custom**, and then enter the range in CIDR notation.

For more information, see *Adding Rules to a Security Group* in the *Amazon EC2 User Guide for Windows Instances* at the AWS website.

**Note:** This port is required to receive incoming unencrypted web traffic.

8. Click **Add Rule**, in the **Type** list, select **HTTPS (443)**, and in the **Source** column, choose **My IP** for each rule. This limits inbound traffic to your computer. To specify an IP address range instead, choose **Custom**, and then enter the range in CIDR notation.

For more information, see *Adding Rules to a Security Group* in the *Amazon EC2 User Guide for Windows Instances* at the AWS website.

**Note:** This port is required to receive incoming encrypted web traffic.

9. Click **Add Rule**, in the **Type** list, select **RDP (3389)**, and in the **Source** column, choose **My IP** for each rule. This limits inbound traffic to your computer. To specify an IP address range instead, choose **Custom**, and then enter the range in CIDR notation.

For more information, see *Adding Rules to a Security Group* in the *Amazon EC2 User Guide for Windows Instances* at the AWS website.

**Note:** This port is required to accept Remote Desktop Protocol (RDP) requests so that you can connect to the EC2 instance.

10. Click **Create**.
Step 3: Launch an Amazon EC2 instance

After you create your VPC, you can launch an Amazon EC2 instance into it.

For more information about how to launch and connect to a Windows instance, see Getting Started with Amazon EC2 Windows Instances in the Amazon EC2 User Guide for Windows Instances at the AWS website.

1. Navigate to the Amazon EC2 console (https://console.aws.amazon.com/ec2/).

2. Use the region selector to choose the location where you created your VPC.

3. Under Create Instance, click Launch Instance.

4. Select an Amazon Machine Image (AMI) that meets the system requirements for Tableau Server.

5. On the Step 2: Choose Instance Type screen, select the instance size you want (for example, m4.2xlarge).

   **Note:** For Tableau Server v10 on a 64-bit virtual machine, you need a minimum of 4 physical cores. On AWS, this means 8 vCPUs. For more information, see Amazon EC2 Instance Types at the AWS website.

6. Click Next: Configure Instance Details.

7. On the Step 3: Configure Instance Details screen, in the Network list, select your VPC.

8. Click through the steps in the wizard until you get to Step 6: Configure Security Group.

9. On the Step 6: Configure Security Group screen, click Select an existing
security group, and then select the security group that you created earlier.

10. Click Review and Launch, review your configuration, and then click Launch.

11. When you are prompted, create a new key pair, download it as a .pem file, and keep it in a safe place. You need the key pair in order to create a password that you can use to log into the Amazon EC2 instance.

12. When you have finished creating and downloading your key pair, click Launch Instances.

13. Click View Instances and locate your instance in the list.

14. Copy the instance ID. You’ll need this in the next step.

Step 4: Create an elastic IP address for the VPC

After you launch your Amazon EC2 instance, you can create and allocate a static public IP address to your VPC.

1. Navigate to the Amazon VPC console (https://console.aws.amazon.com/vpc/).

2. Use the region selector to choose the location where you created your VPC.

3. In the navigation pane, click Elastic IPs.

4. Click Allocate new address, and then click Allocate.

5. In the New address request succeeded dialog box, click the Elastic IP address.

6. On the Action menu, click Associate address.

7. In Resource type, select Instance.

8. In the Instance drop-down list box, select your instance, and then click Associate.

9. Copy the new private IP address. You’ll need this later.
Step 5: Log in to Amazon EC2

To run Tableau Server on AWS, you install it on the Amazon EC2 instance or instances that you just configured. To begin, you log into the Amazon EC2 instance where you want to install Tableau Server.

Use Remote Desktop Protocol (RDP) from a computer running Microsoft Windows to connect to your Amazon EC2 instance running Microsoft Windows. If you’re connecting from a computer running Linux, you can use rdesktop to connect to your Amazon EC2 instance running Microsoft Windows. For more information, see Connecting to Your Windows Instance in the Amazon EC2 User Guide for Windows Instances at the AWS website.

1. Navigate to the Amazon EC2 console (https://console.aws.amazon.com/ec2/).

2. Use the region selector to choose the location where you launched your instance.

3. In the EC2 Dashboard, click Instances.

4. Select the check box next to your instance, and then on the Actions menu, click Connect.

5. In the Connect To Your Instance dialog box, click Download Remote Desktop File to create an .rdp file that’s configured to connect to the instance. The .rdp file is configured to connect to the public IP address of your VPC, which you specified in the Elastic IP address that you created.

6. Click Get Password, and then select the .pem file for the key pair you created earlier. This fills in the private key from your key pair.

7. Click Decrypt Password. When the password is displayed, make a note of the Public DNS address, user name (such as Administrator), and password.

8. Double-click the .rdp file that you downloaded earlier.
9. Click **Connect**. (You can ignore any messages that indicate that the publisher is unknown.)

10. Enter the user name and password that you recorded a few steps ago, and then click **OK**.

11. You can safely ignore any warnings that the publisher of the remote connection is unknown. Click **Yes** to connect.

### Step 6: Install Tableau Server

Now that you're logged in to your EC2 instance, you can install Tableau Server. The installation steps on an EC2 instance are the same as they are for any other computer. For instructions, see **Install and Configure**.

After you install Tableau Server and create a Tableau Server administrator account and sign in, you can **add users**.

If you want to run a Tableau Server cluster, install servers (as workers) on additional Amazon EC2 instances. For more information, see **Self-Deploy a Distributed Tableau Server**.

### Self-Deploy Tableau Server on AWS in a Distributed Environment

If you want to run Tableau Server in a highly available (HA) distributed environment, also known as a cluster, you need to launch three or more Amazon EC2 instances of the same type and capacity to your Amazon Virtual Private Cloud (VPC) and configure them as worker nodes.

The following scenario assumes that you have three Amazon EC2 instances with Tableau Server installed on each instance. One instance is configured as the primary node, and the other two instances are configured as worker nodes.
The following steps show how to install and deploy Tableau Server on a cluster of three Amazon EC2 instances in a highly available and scalable configuration.

**Step 1: Create a Virtual Private Cloud (VPC)**

(Optional) **Step 2: Create an AWS Directory Service for the VPC**

**Step 3: Deploy Three Amazon EC2 Instances**

**Step 4: Install and Configure Tableau Server**

**Step 5: Create a Load Balancer for the Tableau Server Cluster**

**Step 1: Create a Virtual Private Cloud (VPC)**

The following steps assume that you have an Amazon VPC with at least six subnets (three public and three private) in different Availability Zones, as shown below. Be sure to use static IP addresses when setting up your IP addresses. For information about how to create a VPC with public and private subnets, see Scenario 2: VPC with Public and Private Subnets in the Amazon Virtual Private Cloud User Guide at the AWS website.
Launch one EC2 instance into each of the three subnets in your VPC for use as Remote Desktop gateways.

For more information about how to launch and connect to a Windows instance, see Getting Started with Amazon EC2 Windows Instances in the Amazon EC2 User Guide for Windows Instances at the AWS website.

(Optional) Step 2: Create an AWS Directory Service for the VPC

Follow the steps in the Create a Microsoft AD Directory section, in the AWS Directory Service Administration Guide at the AWS website, to create a fully-managed Samba-based
directory in AWS. When you create a directory with Microsoft AD, the AWS Directory Service creates two directory servers and DNS servers. The directory servers are created in different subnets in your Amazon VPC for redundancy, so that your directory remains accessible even if a failure occurs.

Use the two private subnets available within your Amazon VPC to create the Microsoft AD so that you can run Tableau Server across Availability Zones.

Step 3: Deploy Three Amazon EC2 Instances

Deploy three Amazon EC2 instances across three Availability Zones, as shown in the following figure. You’ll use the node you installed previously as the primary server and these two new nodes as workers. All of the instances should be of the same type and capacity.

These Amazon EC2 instances can be auto-joined to the Simple AD domain created in step 2 either via the console or by following the steps in Joining a Windows Instance to an AWS Directory Service Domain in the Amazon EC2 User Guide for Windows Instances at the AWS website.
After you have launched the EC2 instances, connect to them from one of the Remote Desktop Gateway (RDGW) instances by using the credentials that you decrypted for the local administrator account.

**Step 4: Install and Configure Tableau Server**

You’ll install Tableau Server on the Amazon EC2 instances you launched in Step 3: Deploy Three Amazon EC2 Instances and configure the instances as primary and worker servers. For more information about installing and configuring Tableau Server on a primary and
worker nodes, see Quick Start: Distributed Server.

Step 5: Create a Load Balancer for the Tableau Server Cluster

Follow the steps outlined in Getting Started with Elastic Load Balancing in the Elastic Load Balancing User Guide at the AWS website to launch a load balancer within your VPC.

1. In Step 1: Select a Load Balancer Type, if you want the load balancer to be publicly accessible, select the two public subnets. Otherwise, select the Create an internal load balancer check box and choose the two private subnets.

If you choose to expose the load balancer with a public endpoint, make sure that you configure Elastic Load Balancing with SSL, as explained in Create a Classic Load Balancer with an HTTPS Listener in the Elastic Load Balancing Classic Load Balancer Guide at the AWS website.

2. In Step 2: Configure Your Load Balancer and Listener of the instructions, in the Elastic Load Balancing User Guide at the AWS website, ensure that your security group is configured to allow access on port 80 or 443 only, with the source limited to hosts or ranges of hosts that will access Tableau Server.

3. In Step 4: Configure Your Target Group, you can specify the ping path as /.

4. In Step 5: Register Targets with Your Target Group, select the Tableau Server instances and ensure that Enable CrossZone Load Balancing is selected so that the load balancer can load-balance the traffic across the instances in multiple Availability Zones.

5. Update Tableau Server to use the load balancer. For more information, see Add a Load Balancer in the Tableau Server Help.

Alternatively, you can configure Tableau Server to work with a load balancer by performing
the following steps.

1. Create a subnet for your load balancer that has a CIDR block with a /27 bitmask, for 32 IP addresses.

   **Note:** The IP addresses provided by Elastic Load Balancing are dynamic, and Tableau Server needs a list of static IP addresses for this configuration. To make this work, we’re creating a subnet with the smallest possible CIDR range so that the IP addresses the load balancer has are limited to a finite set.

For the next steps we’ll use the tabadmin command. This command enables you to perform administrative tasks from the command line on Tableau Server and installs with Tableau Server by default. For a general overview, see How to Use tabadmin in the Tableau Server Help.

2. In the Tableau Server bin directory, enter the following command, where name is the URL that will be used to reach Tableau Server through the load balancer:

   ```bash
   tabadmin set gateway.public.host "name"
   ```

3. Enter the following command, where server1, server2, and so on are the IP addresses for the given CIDR range of subnets for Elastic Load Balancing:

   ```bash
   tabadmin set gateway.trusted "server1,server2,..,server30"
   ```

4. Apply the changes:

   ```bash
   tabadmin config
   ```

5. Start the server so the changes can take effect.

   ```bash
   tabadmin start
   ```
Securing Tableau Server on AWS

Whether you deploy Tableau Server on premises or in the cloud, it’s important to take steps to make your deployment secure. For information about making Tableau Server more secure, see Security.

In addition to the security features built in to Tableau Server, AWS provides other features that you can use to help secure your Tableau Server environment, such as:

- **Amazon VPC** adds another layer of network security to your environment by creating private subnets.

- **Security Groups** determine which inbound and outbound traffic can connect to your network. Limit inbound to your IP addresses in your Classless Inter-Domain Routing (CIDR) block. Do not use 0000\0, which is unsecure because it allows all traffic to access your server.

- **AWS Identity and Access Management (IAM)** allows specific control over user access to features within AWS.

- **AWS Direct Connect** allows a dedicated network connection from a corporate network to AWS using industry-standard 802.1Q VLANs through an AWS Direct Connect partner. For more information, see Requesting Cross Connects at AWS Direct Connect Locations in the AWS Direct Connect User Guide at the AWS website.

- **Amazon EBS Encryption** offers a simple and performant way to encrypt data at rest inside your disk volumes and data-in-transit between EC2 instances and EBS storage.

You can implement enterprise application security in AWS and Tableau Server to enable a single report or dashboard to securely serve the needs of a broad and diverse user base, including both internal and external users. Enterprise application security has three main components:
Network

Network security for Tableau Server in AWS relies on the use of Amazon VPC security groups with SSL for securing internal and external communications. For more information, see Security Groups for Your VPC in the Amazon Virtual Private Cloud User Guide at the AWS website.

Amazon VPC

An Amazon VPC is a distinct, isolated network within the cloud; network traffic within each Amazon VPC is isolated from all other Amazon VPCs. Using an Amazon VPC allows you to create your own network subnets and divide application layers into network subnets for a greater level of control. We recommend that you install and run Tableau Server in a separate subnet within your Amazon VPC so that you can configure the network for access to Tableau Server and other data sets. The following figure shows a typical installation of a single-node Tableau Server in an Amazon VPC.
Security Groups

Security groups enable you to define what types of network traffic can access Tableau Server. Amazon EC2 security groups act as a firewall that governs network traffic into and out of Amazon EC2 instances. You can define and assign security groups that are appropriate for your Amazon EC2 instances. By default, Amazon EC2 instances are launched with security groups that allow no inbound traffic. Before you can access your EC2 instance, you need to make changes to allow the appropriate inbound traffic.

Here are the minimum requirements for connections to Tableau Server on an EC2 instance:

- Connection via RDP (port 3389) using a Remote Desktop client to access and manage the instance and services.
- Standard web traffic via HTTP (port 80) and HTTPS (port 443), to view content hosted on, and to publish to Tableau Server.
- Communication between Tableau Server components on different instances (if any) should be allowed. See the ports listed under All and Distributed/High Availability categories.

Based on these requirements, you should enable only three standard ports for inbound traffic to your EC2 instance: HTTP 80, HTTPS 443, and RDP 3389. You should also limit remote access (port 3389) from a few hosts, and also limit HTTP and HTTPS traffic to hosts within your corporate network or to a trusted set of clients.

Client Access

By default, Tableau Server uses standard HTTP requests and responses. Tableau Server can be configured for HTTPS (SSL) with customer-supplied security certificates. When Tableau Server is configured for SSL, all content and communications between clients are encrypted and use the HTTPS protocol. When you configure Tableau Server for SSL, the browser and SSL library on the server negotiate a common encryption level. Tableau Server uses OpenSSL as the server-side SSL library, and is pre-configured to use currently
accepted standards. Each web browser that accesses Tableau Server via SSL uses the standard SSL implementation provided by that browser. For more information about how Tableau Server uses SSL, see SSL. Tableau Server will listen for SSL traffic only on port 443. You may not configure custom ports for SSL/TLS.

If you’re using Elastic Load Balancing (ELB), ELB can also perform SSL termination on your behalf. Allowing ELB to handle encryption/decryption of web traffic is an easy way to secure the client’s connection with Tableau Server without needing to manually configure SSL on Tableau Server itself. For more information, see AWS Elastic Load Balancing: Support for SSL Termination at the AWS website.

AWS Directory Service

Optional. The AWS Directory Service is a managed service that allows you to connect your AWS resources to an existing on-premises directory such as Microsoft Active Directory (with AD Connector), or to set up a new, stand-alone directory in the AWS cloud (with Simple AD). Connecting to an on-premises directory is easy, and after this connection is established, all users can access AWS resources and applications with their existing corporate credentials.

Using the AWS Directory Service, you can choose to use Active Directory-based authentication instead of local authentication, which creates users and assigns passwords using Tableau Server’s built-in user management system. To set up Active Directory-based authentication, in the configuration step after installing Tableau Server, you must choose Active Directory. It is not possible to switch between Active Directory and local authentication later.

The Active Directory authentication model uses the Microsoft Security Support Provider Interface (SSPI) to sign in your users automatically, based on their Windows user name and password. This creates an experience similar to single sign-on (SSO).

Data

Tableau Server uses native drivers (relying on a generic ODBC adapter when native drivers are not available) to connect to databases whenever possible, for processing result sets,
refreshing extracts, and for all other communications with the database. You can configure the driver to communicate on non-standard ports or use transport encryption, but this type of configuration is transparent to Tableau Server. However, since the Tableau Server-to-database communication is typically behind a firewall, you may choose not to encrypt this communication.

Connecting to Data Stores in AWS

You can launch AWS resources, such as Amazon Relational Database Service (Amazon RDS), Amazon Elastic MapReduce (Amazon EMR), or Amazon Redshift, into an Amazon VPC. By placing the Tableau Server into the same Amazon VPC as your data stores, you can ensure that your traffic never leaves the Amazon VPC.

You can use subnets with security groups to launch your resources into different layers but allow them to communicate securely within an Amazon VPC, as illustrated in the following diagram.
Connecting to Data Stores Outside of AWS

You can optionally connect your Amazon VPC to your own corporate data center by using an IPsec hardware VPN connection, thus making the AWS cloud an extension of your data center. A VPN connection consists of a virtual private gateway attached to your Amazon VPC and a customer gateway located in your data center. You might choose to use AWS Direct Connect, which is a network service that provides an alternative to using the Internet to utilize AWS cloud services. AWS Direct Connect lets you establish a dedicated network connection by using industry-standard 802.1Q VLANs through an AWS Direct Connect partner. For more information, see Requesting Cross Connects at AWS Direct Connect Locations in the AWS Direct Connect User Guide at the AWS website.

You can use the same connection to access public resources (such as objects stored in Amazon Simple Storage Service (Amazon S3) using public IP address space) and private resources (such as Amazon EC2 instances running within an Amazon VPC using a private IP space), while maintaining network separation between the public and private environments.

Encrypting Data at Rest

Amazon EBS encryption offers a transparent and simple way to encrypt volumes which may contain personally identifiable information (PII). EBS encryption encrypts both data at rest inside the volume and data in transit between the volume and the instance using AES-256. This feature has little-to-no impact on Tableau Server performance. Therefore, we recommend that you take advantage of this service regardless of whether your systems store PII.

Optimizing Performance of Tableau Server on AWS

Optimizing the performance of Tableau Server when it is installed on an Amazon EC2 instance in the AWS cloud adds another dimension to tuning your Tableau Server solution. This section discusses tuning Tableau Server for the cloud. For general performance tuning information, see the Tableau Server Performance Overview. For information about tools you can use to help optimize performance, see Performance Resources.
It is important to keep in mind that every deployment of Tableau Server on AWS is different because everyone’s workload is unique. Your company’s employees are different, leverage different data, ask different kinds of questions, and have different business needs than other companies. For this reason, we recommend testing your Tableau Server workloads on a variety of different Amazon EC2 instance types before putting them into production. The demands of your workload will likely be influenced by the following factors:

- Heavy, light, or moderate Tableau Data Extract use
- The proportion of people who view vs. interact with visualizations and dashboards
- Tableau Data Extract refreshes which occur during or after working hours
- The number of concurrent individuals during any given period
- View and dashboard complexity
- Size of the community who leverage Tableau Web Authoring

There are some general guidelines that you can follow to increase the likelihood of choosing the right instance type. Using TabJolt, a free scalability testing tool built by the Tableau, we ran 1700 load tests across 34 Amazon EC2 Instances to test for performance and scalability using the following metrics:

- Number of virtual users executing
- Average Transactions per Second
- Average Success Response Time
- Average Error Rate (the test considers any viz that takes > 60 seconds to render an error)

Performance Best Practices

The following performance best practices may be useful for you to reference as you deploy Tableau on AWS:
• **Always run at least 8 cores per Amazon EC2 Instance**

Even with a relatively low number of users, EC2 Instances with fewer than 16 vCPUs (the equivalent of 8 cores), do not consistently perform well. For example, one r4.4xlarge instance, which has 16vCPUs, is able to handle more users, with a lower response time and error rate, than two r4.2xlarge instances, which have 8vCPUs each. This pattern is consistent as you scale up, with four 16vCPU instances and two 32vCPU instances dramatically outperforming eight 8vCPU instances.

• **The workload greatly determines the results**

The robustness of your workload goes a long way in determining how it performs on different EC2 instance types. Using a different set of dashboards, for example, you will see noticeable differences in performance from the same underlying virtual machine instances. Attempting to compare performance using any other workload than your own is not very useful.

• **More CPU is better**

CPU is often the main bottleneck for Tableau Server performance. In general, when you want to do more with Tableau, we recommend you add more and better CPU.

• **Make sure your Amazon EC2 instance has enough RAM**

When we ran the same workloads on instances with less CPU but more RAM, we experienced higher transactions per second (TPS), lower response time, and lower error rate. EC2 instances with too little RAM can cancel out the benefit of high-end CPU. In production, run with at least 30 GB RAM but shoot for 8 GB of RAM per core. While it is important to choose an instance with a lot of CPU, running Tableau Server on instances starved for RAM is going to lead to low performance—no matter how much CPU you have.

• **You want to use SSD-based volumes, but may not need Provisioned IOPS**
Tableau Server encompasses a number of processes and components, including an industrial strength database (PostgreSQL) that stores the system's metadata. Tableau Server needs a reasonable level of disk throughput in order to perform well, and we advise using Amazon Elastic Block Store (EBS) SSD-based volumes only. Magnetic disks lack the throughput needed to handle the database's requests effectively. In our test, we ran both general purpose SSD (gp2) and EBS-provisioned IOPS volumes, with two EBS disks being used for most tests. Most EBS volumes had 1500 provisioned IOPS. After running the tests again with General Purpose SSDs, our results were nearly identical with moderately demanding workloads. While there are certainly cases where Provisioned IOPS will make a noticeable difference in the performance of your Tableau Server workloads on AWS, you shouldn’t assume that you need Provisioned IOPS by default. The best way to find out, of course, is to test your Tableau Server workloads yourself.

- Do your own testing with TabJolt

Your workload and the configuration of your Amazon EC2 Instances can make a major difference in how Tableau Server performs for you. The flexibility of EC2 makes it easy to determine combination of settings and instance types to suit your needs. For example, isolating a few RAM-and-CPU hungry processes on several machines instead of running them on all machines in your cluster makes a huge difference in transactions per second (TPS). Ignore the temptation to compare performance characteristics of different workloads. While this is admittedly fun to do, it isn’t very useful. Instead, use the same workload and tinker with your hardware and software configuration for best effect. AWS makes this very easy to do.

**Scaling Tableau Server on AWS**

Tableau Server is designed to scale up with more CPU cores and memory, and scale out when you add servers. This architecture allows you to maximize the use of compute resources while giving you the ability to scale massively.
To build in redundancy, you need to add additional servers, which host copies of the repository, file store, and other processes. In this scenario, you should isolate the initial server in its own node, and it should ideally run as few of the server processes as possible.

**Note:** Although you can split an 8-core license onto two 4-core machines, we recommend that you scale in increments of 8-core machines only.

**Load Balancing**

On AWS, Elastic Load Balancing (ELB) automatically distributes incoming application traffic across multiple Amazon EC2 instances in the cloud. It enables you to achieve greater levels of fault tolerance in your applications, and seamlessly provides the required amount of load balancing capacity needed to distribute application traffic.

You can use ELB to distribute requests across multiple gateways in a Tableau Server cluster. In the figure below, all three nodes have gateways, which are used to route requests to available server processes. Unlike the repository process, there aren’t any passive or standby gateway processes—all gateways are active. When you add a load balancer to a Tableau Server cluster, the URL that’s accessed by Tableau Server users belongs to the load balancer, not to the initial Tableau Server.
High Availability

Now that you have provided redundancy for the file store, repository, and gateway by adding additional nodes, you can additionally build redundancy for the initial Tableau Server. You can do this by creating a backup of the initial Tableau Server. Although the backup must be licensed during installation, it does not count as one of the three environments allowable under the Tableau end user license agreement (EULA).
To configure for high availability, you need to run a failover cluster plus an additional computer as the backup for your primary Tableau Server. If you configure for high availability, the initial Tableau Server and the backup may be running few or no Tableau Server processes. For more information, see High Availability in the Tableau Server Help.

Troubleshooting Tableau Server on AWS

Follow the suggestions in this topic to resolve common issues with Tableau Server when installed on an Amazon EC2 instance in the AWS cloud.

- **Tableau Server does not use all CPU cores**

  One or more of the following symptoms might occur when working with Tableau Server core-based licensing:

  - **When installing Tableau Server**, the installer might show fewer cores than you would expect to see for the computer running Tableau Server.
  
  - **On the Licenses page on Tableau Server**, the number of cores listed under Licenses in Use is less than you would expect to see.

  - **When you run `tabadmin licenses`, Tableau Server might report fewer cores in use that you would expect to see.**

    For more information, see Not All Cores Recognized by Tableau Server.

- **There is high I/O latency**

  Changing the disk driver performance setting to Better Performance may give you better results.
Manage Sites

What is a site .................................................................................................................. 422
Planning a Site .............................................................................................................. 425
Manage Users and Groups ............................................................................................ 428
Dashboard-based Custom Portals .................................................................................. 478
Data Connection Authentication ..................................................................................... 482
Manage Content Access .................................................................................................. 508

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.

Note: This article pertains to configuring sites on Tableau Server deployments. For Tableau Online, see Site Administrator Role and Tasks.
In this article

Site administrator tasks
Steps for setting up your site
See also

Site administrator tasks

Where the Server Administrator site role gives a user unrestricted access to the entire Tableau Server deployment, the Site Administrator site role gives a user unrestricted access at the site level.

Although a server administrator can work at both the server and site levels, we make a distinction between the two sets of tasks. 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. Site administrator tasks include any of the following:

- 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>
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.

Once 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.

See also

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.

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
- Site roles and permissions
- Extract refresh schedules

- Steps for setting up your site

The subsequent sections go over these site components, assuming that you are familiar with

**Note:** This article and section apply only to Tableau Server deployments that you maintain, whether 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 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 data sources and workbooks to a site must be able to sign in to it. 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. Tableau Server licenses are based on either cores or users. If the server has a user-based license, there’s an upper limit to how many users can have active accounts on the server. 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.

As we just noted, site roles act as an upper limit on permissions. Permissions determine what a user can do within the context of the site role. A user whose site role is Interactor can never publish to the site, regardless of the permissions you grant them. A user whose site role is Publisher 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**

**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 can add users to sites in the following ways:

- By adding a local user account or a user account from Active Directory, as described in this topic. You can also add users by importing an Active Directory group. For details, see Create Groups via Active Directory.

- By importing a CSV file that contains user information. For details, see Import Users and CSV Import File Guidelines.

In a single-site environment, administrators can add users to a site on the Users page. In a
multi-site environment, you will use the Site Users page. Server administrators must give site administrators the ability to add users to sites. This setting can be enabled or disabled by the server administrator.

**Note:** Users can be added to sites, or to the server. To add users to the server, see Add Users to the Server. The options available for adding users depends on the authentication method that you select when you first configure Tableau Server. If you are using local authentication, you cannot add Active Directory users. If you are using Active Directory, you cannot add local users.

If you add a user without assigning site membership and role, the user is assigned the Unlicensed role and won’t use a server license (user-based licensing only). The user will exist in Server Users, but will not be a member of any site until you add that user to the site.

On the **Users** (single-site) or **Site Users** (multi-site) page you can see the users on the site you’re currently signed into. You can add users to (or remove them from) the current site only. If a user belongs to more than one site, you can remove that user from the current site.

**Note:** When a site administrator removes a user from a site (and the user only belongs to that one site), the user will be automatically deleted from the server if that user doesn’t own any content.
This screenshot is from a single-site environment. In a multi-site environment, this would be the Site Users page.

To add local users to a site

1. In a site, click **Users**, click **Add Users**, and then click **New User**.

2. Enter a user name. If the server is configured for local authentication, using an email
address for the user name is the best way to avoid user name collisions (for example, jsmith@example.com instead of jsmith).

Also enter information in the following fields:

- **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.

3. Select a site role. For details on site roles, see Set Users’ Site Roles.

4. Click **Add User**.

**Note for multi-site servers:** A site administrator can edit an existing local user account only if the administrator has control over 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 site B only cannot edit User1’s full name or reset the password.

To 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). 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 details on site roles, see Set Users’ Site Roles.

4. Click Import Users.

Set Users’ Site Roles

When you add users to a site on Tableau Server, you must apply a site role to them. The site role determines which users or groups can publish, interact with, or only view published content, or who can manage the site’s users and administer the site itself.

Note: Tableau Server site roles do not correspond to user licenses that you purchase from Tableau (if you are using user-based licensing instead of core-based server licensing). Those licenses allow a certain number of users on the server.
Looking for Tableau Server on Linux? See Site User’s Site Roles.

In this article

Who can publish content

Capabilities the site role determines

General capabilities allowed with each site role

Change a user’s site role

Site roles and Active Directory import and synchronization

Who can publish content

The following site roles allow publishing content:

- Server Administrator (Tableau Server only)
- Site Administrator
- Publisher

Users with a site role of Interactor, Viewer, and Unlicensed cannot publish content to the server.

Capabilities the site role determines

The site role is set at the user level (that is, not on content), to cap what a user can do with content site-wide. For example, a person whose site role is Interactor can never publish content from Tableau Desktop, even if a project’s permission rule assigns the Publisher permissions role to a group the user is a member of.

Site roles also determine who can configure the site or server itself. For example only a Site Administrator or Server Administrator can manage users.
General capabilities allowed with each site role

- **Server Administrator** (Tableau Server only): The server administrator has full access to all server and site functionality, all content on the server, and all users.

- **Site Administrator**: Site administrators can manage groups, projects, workbooks, and data sources (including connection information) for the site.

  On Tableau Server, server administrators can determine whether to not 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.

  Site administrators also have unrestricted access to the site’s content. A user can be a site administrator on multiple sites.

- **Publisher**: Publishers can sign in, browse the server, and interact with the published views. They also can connect to Tableau Server from Tableau Desktop to publish (upload) and download workbooks and data sources. Publishers cannot manage users.

- **Interactor**: Interactors can sign in, browse the server, and interact with the published views. They are not allowed to publish to the server.

- **Viewer**: Viewers can sign in and see published views on the server but cannot use any interaction features like filtering and sorting. Users with this site role can be allowed only to view and add and view comments.

- **Unlicensed**: Unlicensed users cannot sign in to the server. Users are assigned the Unlicensed role in the following circumstances:

  - You import users from a CSV file.
  
  - 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.

See also How Permissions are Evaluated.

Change a user’s site role

By default both server and site administrators can manage user settings. On Tableau Server, server administrators can configure a site to disallow site administrators the ability to manage users.

1. Sign in to the site, and go to the **Users** area.

2. Select the users, and then select **Actions > Site Role**.

3. Select the new site role, and then click **Change Site Role**.
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.

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 users and groups will promote a user's site role, but not demote a user's site role.

If a user already has the ability to publish, that ability is maintained.

To guarantee a user maintains a site role with equal or greater capabilities in server after an import, the following matrix 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
- Publisher: P
- Interactor: I
- Unlicensed: U

<table>
<thead>
<tr>
<th>Import Site Role</th>
<th>Current Site Role</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>SA</td>
</tr>
<tr>
<td>Site Administrator</td>
<td>SA</td>
</tr>
<tr>
<td>(SA)</td>
<td></td>
</tr>
<tr>
<td>Publisher</td>
<td>SA</td>
</tr>
<tr>
<td>(P)</td>
<td></td>
</tr>
<tr>
<td>Interactor</td>
<td>SA</td>
</tr>
<tr>
<td>Import Site Role</td>
<td>Current Site Role</td>
</tr>
<tr>
<td>-----------------</td>
<td>------------------</td>
</tr>
<tr>
<td>(I)</td>
<td>SA P I V U</td>
</tr>
<tr>
<td>Viewer (V)</td>
<td>SA P I V V</td>
</tr>
<tr>
<td>Unlicensed (U)</td>
<td>SA P I V U</td>
</tr>
</tbody>
</table>

### Assign Site Membership

You can set a site so that both server and site administrators can add or remove users on that site and change their site roles. At the All Sites level, server administrators can manage the sites existing users can sign in to (site membership), and their site role on each.

1. In the site menu, click **Manage All Sites**, and then click **Users**.

2. Select one or more users, and then select **Actions > Site Membership**.
3. Select one or more sites, and a role for each site, and then click **Save**.
Guest User

Core-based licenses of Tableau Server include a Guest user option, which you can use to allow people who don’t have an account on the server to see and interact with Tableau views embedded in web pages.

Guest user access is enabled by default when Tableau Server is installed with a core-based license. If you do not intend to use Guest user access, then you should disable it.

Guest access allows users only 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.

**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.

Guest user permissions

A Guest user can have the following permissions:

- **Projects, 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

When a Guest user is included in a group that has a permission rule set on a content item, Guest user permissions do not affect the permission-levels of other users in that group.

Enable or Disable Guest access

You must be a server administrator to change Guest account settings at either the server or the site level. Guest access is enabled by default in core-based licensing installation.
1. In the site menu, click **Manage All Sites** and then click **Settings > General**.

2. For **Guest Access**, select or clear **Enable Guest account** to toggle Guest user 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 Settings page, clear the **Enable Guest account** 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.

**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 embedded Tableau views.

- You cannot edit it or select it as the owner of a content resource.

- You can add it as a member of groups on a site.

- When enabled, it is a member of the All Users group.

- You cannot delete the account; however, you can turn off access to it by clearing the check box described in the steps above.

- 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.
- It is not allowed to save custom views.
- Guest cannot be used in a user filter.

**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 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**.
   
   ![Add Users to this Site](image)
   
   - 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**.

   ![Import users from file](image)

   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, you can include site-role assignments 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 only if it gives the user more access. Importing users and groups will promote a user’s site role but not demote it.

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**.

When you assign site membership to a user, you can specify the user’s site role for each site. For information, see Assign 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.

Note: The issue of credentials mismatch during import doesn't apply if the server is configured to use Active Directory for authentication. In that case, the CSV file should never contain a password, because user passwords are managed by Active Directory.

If you’re importing users to a site and think that the users might already exist on the server, you can try leaving the Password column in the CSV file blank. When you import the users, 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.

**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 site role and the ability to publish, 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. For details, see Import Users or createsiteusers filename.csv.

You can import users at the site or server level. If you import users to a site, site roles are applied to the user. If you specify site roles, but importing users would exceed your license limits, users are imported as Unlicensed. If you import users to the server (not into a specific site), the user isn’t assigned to a site, and site roles in the CSV file, such as Publisher and Interactor, are ignored, and the Unlicensed site role is assigned during the import.

**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 name includes the "@" character other than as a domain separator, you need to refer to the symbol using the hex format: `\0x40`

  For example, `user@fremont@myco.com` should be `user-\0x40fremont@myco.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 that's 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.

- Site role. The site role determines the maximum permissions a user can have on content published to the site. Site roles include Site Administrator, Publisher, Interactor, or Viewer, and a couple of others for publishing only. For more information, see Set Users' Site Roles.
Note: Tableau versions earlier than 9.0 use license level instead of site roles. For more information, go to the Tableau Help page and select the documentation specific to your version.

- Administrator level (System, Site, or None). This setting determines whether the user is imported as an administrator. If you are using the site administration pages, you can set the administrator role to System only if you are importing while managing the server. If you are using the server administration pages to import users while you are managing a site, and if the administrator role for a user in the CSV file is set to System, Tableau Server imports the user as a site administrator.

- Publisher permissions (yes/true/1 or no/false/0). This setting determines whether the user has publisher permissions. If you are using the site administration pages, the publisher setting is used only if you are importing into an individual site. If you are importing users while managing a server, this value isn’t used.

- Email address. The email address is part of the information that’s 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, etc., regardless of the content in the columns.

Improve performance for large CSV files passed through tabcmd

A server administrator can use the tabadmin set command to enable settings that help to improve performance for importing large CSV files through tabcmd commands. 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 memory required to process the file. These tabadmin set options apply to the tab-
cmd createsiteusers, deletesiteusers, addusers, and removeusers commands.

- vizportal.csv_user_mgmt.index_site_users
- vizportal.csv_user_mgmt.bulk_index_users
- searcheserver.index.bulk_query_user_groups

You can find descriptions for these settings in tabadmin configuration options.

Settings and site roles

The license level, administrator, and publisher settings for a user are used during the import process to set a user's site role. 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>
<tbody>
<tr>
<td>License level=(any) Administrator=System Publisher=true</td>
<td>System (server) administrator. This setting is valid only if you are importing users while managing the server.</td>
</tr>
<tr>
<td>License level=(any) Administrator=Site Publisher=true</td>
<td>Site administrator. This setting is valid only if you are importing users while managing a specific site.</td>
</tr>
<tr>
<td>License level=Interactor Administrator=None Publisher=true</td>
<td>Publisher</td>
</tr>
<tr>
<td>License level=Interactor Administrator=None Publisher=false</td>
<td>Interactor</td>
</tr>
<tr>
<td>License level=Viewer Administrator=None Publisher=false</td>
<td>Viewer</td>
</tr>
</tbody>
</table>
CSV settings | Site role
---|---
License level=Unlicensed Administrator=None Publisher=false | Unlicensed

Notes

- If you are importing users while managing the server, you can create users with only two site roles: system (server) administrator and Unlicensed. All other settings are site specific. In that case, if the administrator level for a user in the CSV file is not System, the user’s site role is set to Unlicensed.

- 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.

Example

The following example shows a CSV file that contains information for several users.

```plaintext
henryw,passw0rd,Henry Wilson,Interactor,None,yes,henryw@example.com
freds,pa$$word,Fred Suzuki,Viewer,None,no,freds@example.com
alanw,p@ssword,Alan Wang,Interactor,Site,yes,alanw@example.com
michellek,mypassword,Michelle Kim,Interactor,System,yes,michellek@example.com
```

If you import this file while managing a site, four users are added to that site. The Administrator mode for user Michelle is set to System. However, because you are importing the users into a site, Tableau Server sets user Michelle to be a site administrator, not a system administrator. 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. The site roles in the CSV file (Interactor and Viewer) must be associated with site users, so the site role for the users who are not administrators is set to Unlicensed.
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.
View, Edit, and Delete Users

View and edit site users

Sign in to a site as an administrator, and then click Users. In this page you can set group membership, set site role, or remove the user from the site.

Click a user name to see the content they own.
Click **Settings** in a user page to view their account settings. The user **Settings** page is available when the user is a member only of sites that the site administrator also controls, and site administrators are allowed to manage users in the site settings.

If Tableau Server is running multiple sites, **Server Users** lists all users on the server system, and **Site Users** displays all users for the current site.

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.

**For multi-site servers:** Site administrators can edit an existing user’s account as long as the user is a member only of sites that the site administrator also controls, and site administrators are allowed to manage users in the site settings. For example, if User
Joe is a member of Site A and Site B and the site administrator is only an administrator of Site B, the site administrator cannot edit Joe’s Full Name or reset his password.

View and edit server users

Sign into Tableau Server as a server administrator. On the site menu, click Manage All Sites, and then click Users. In this page you can set site membership or delete the user from the server.

Click a user name to view account settings. The user Settings page is available when the user is a member only of sites that the site administrator also controls, and site administrators are allowed to manage users in the site settings.
Search for users

To search for a specific user, in the **Search** box on the left, type all or part of the user’s name, and then press **Enter**.

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 from a site 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 not removed.

Note: When a site administrator removes a user from a site (and the user only belongs to that one site), the user will be automatically deleted from the server if that user doesn’t own any content.

1. In a site, click **Users**. Select one or more users to delete, and then select **Actions > Remove**.

2. Click **Remove** in the confirmation dialog.

Delete users from the server

You can delete a user from Tableau Server only if the user does not own any content (projects, workbooks, views, or data sources). If you attempt to delete a user who owns content,
the user site role will be set to Unlicensed, but the user will not be deleted.

If a user is a member of multiple sites, and owns content in one or more of those sites, the user will be removed from the sites in which they don’t own content. The user will remain a member in sites where they do 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 Password**.

---

**Groups**

### 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.

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**.

![New Group dialog box](image)

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.

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.

1. In a site, click **Groups**.

   On the Groups page, select one or more groups.
2. Click **Actions > Synchronize.**

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 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.

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 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.

To change this behavior run the following commands in tabadmin:

```
tabadmin stop
tabadmin set vizportal.adsync.update_system_user true
tabadmin configure
tabadmin start
```

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 on a regular schedule or on-demand on the General tab of the Settings page for the server.
The Last synchronized time indicates the time that synchronization most recently began.

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.

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.

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.

2 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.

3 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](image)

**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 provid-
ing 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>
<tr>
<td>Facilities</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Hospital occupancy</td>
</tr>
<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.
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`

Data Connection Authentication

More information

- [Setting an Oracle Connection to Use TNSNames.ora or LDAP.ora](Tableau Support) (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.

Tableau Server requires constrained delegation, with the Run As User account specifically granted delegation rights to the target database Service Principal Names (SPNs). Delegation is not enabled in Active Directory by default.

Note: Kerberos constrained delegation for SSO to Tableau Server is not supported. Constrained delegation for data sources is supported. The client computer must have a
TGT from the Active Directory domain. For more information, see Single-Sign On (SSO) in Kerberos Requirements.

To configure Kerberos delegation:


2. On all nodes in Tableau Server, configure the Run As User to act as part of the operating system. For more information, see Enable Run As User to Act as the Operating System.

3. In Active Directory:
   - Configure SPNs for the data sources you will be using.
   - Enable Kerberos delegation for the data sources’ SPNs
   - (Optional for multi-domain environments) Configure krb5_conf.html to map principal names to local user names for each Kerberos realm. See Kerberos delegation multi-domain configuration.

4. Enable delegation for data connections:
   - **SQL Server**—See Enabling Kerberos Delegation for SQL Server in the Tableau Community.
   - **MSAS**—See Enabling Kerberos Delegation for MSAS in the Tableau Community.
   - **Oracle**—See Enable Kerberos Delegation for Oracle in the Tableau Community.
   - **Hive/Impala**—See Enable Kerberos Delegation for Hive/Impala in the Tableau Community.
• **PostgreSQL**—See Enabling Kerberos Delegation for PostgreSQL in the Tableau Community.

• **Teradata**—See Enabling Kerberos Delegation for Teradata in the Tableau Community.

### Enable Run As User to Act as the Operating System

To use Kerberos delegation with Tableau Server, you must configure the Run As User account to act as the operating system on each Tableau Server node.

1. On the computer that is running Tableau Server, select **Start > Control Panel > Administrative Tools > Local Security Policy**.

2. In the Local Security Settings window, expand **Local Policies**, click **User Rights Assignments**, and then right-click **Act as part of the operating system** and select **Properties**.
3. In the Act as part of the operating system Properties window, click **Add User or Group**.

4. Type the `<domain>\<username>` for the Tableau Server Run As User account (for example: `MYCOMPANY\tableau_server`), and then click **Check Names**.

5. When the account resolves correctly, it is underlined. Click **OK**.

6. Click **OK** to close the Local Security Policy windows.

**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 Server to access the data.

4. Upon your confirmation, the data provider sends an access token back to Tableau
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.
- 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 two procedures:

- Get required information from Google and enable API access.
- Use the information you obtained to configure your server.

**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**
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.

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:

1. On the Tableau Server computer, open the Command Prompt as an administrator and change to the Tableau Server bin directory.

   ```
cd C:\Program Files\Tableau\Tableau Server\<version>\bin
   ```

2. Type the following command to stop the server:
tabadmin stop

3. Type the following commands to configure the server with the client ID and client secret you obtained from Google, as well as your server URI. Press Enter after each command.

    tabadmin set oauth.google.client_id <your_client_ID>
    
    tabadmin set oauth.google.client_secret <your_client_secret>
    
    tabadmin set oauth.google.redirect_uri <your_authorized_redirect_URI>

4. Type the following commands to complete the configuration and restart the server:

    tabadmin config
    
    tabadmin start

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 pre-requisites 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)

![Available OAuth Scopes and Selected OAuth Scopes](image)

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, open the Command Prompt as an administrator and change to the Tableau Server bin directory:

   ```bash
   cd C:\Program Files\Tableau\Tableau Server\<version>\bin
   ```

2. Type the following command to stop the server:
tabadmin stop

3. Type the following commands to configure the server with the consumer ID and secret you obtained from Salesforce and the callback URL. Press Enter after each command:

```
tabadmin set oauth.salesforce.client_id <your_consumer_ID>
```
```
tabadmin set oauth.salesforce.client_secret <your_consumer_secret>
```
```
tabadmin set oauth.salesforce.redirect_uri <your_callback_URL>
```

4. (Optional) To change the default login server, type the following command:

```
tabadmin set oauth.salesforce.server_base_url <URL>
```

   By default, this is set to `https://login.salesforce.com`.

5. Type the following commands to complete the configuration and restart the server:

```
tabadmin config
```
```
tabadmin start
```

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.
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

1. On the Tableau Server computer, open a command prompt as an administrator and change to the Tableau Server bin directory using the following command:

   ```
   cd C:\Program Files\Tableau\Tableau Server\<version>\bin
   ```

2. Type the following command to stop the server:

   ```
   tabadmin stop
   ```

3. Type the following commands to configure the server with the app token, consumer
key, and consumer secret that you copied from the Intuit site.

`tabadmin set oauth.quickbooks.oauth_callback_uri http://YOUR-SERVER/auth/add_oauth_token`

`tabadmin set oauth.quickbooks.consumer_key <your_consumer_key>`

`tabadmin set oauth.quickbooks.consumer_secret <your_consumer_secret>`

4. Type the following commands to complete the configuration and restart the server:

`tabadmin config`

`tabadmin start`

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 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.

**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. **Single-site**: Click **Settings > General**.

   **Multisite**: In the site menu, click **Manage All Sites** and then click **Settings > General**.

3. In the **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**

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.
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:

- 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

Configure SAP HANA SSO

You can configure Tableau Server to use SAML delegation to provide Single Sign-on (SSO) for SAP HANA. HANA SSO is not dependent on SAML authentication to Tableau Server. **Note:** You do not need to use SAML sign on with Tableau Server in order to use HANA SSO. You can sign in to Tableau Server using whatever method you choose.

With SSO for SAP HANA, Tableau Server functions as an Identity Provider (IdP) and this configuration allows you to provide a single sign-on experience for users making SAP HANA connections. As part of the configuration, you need to acquire a SAML certificate and key file for Tableau Server (these should be a public key certificate and private key). You
need to also install the signed certificate in HANA. You can generate the certificate and key yourself, or get them from a Certificate Authority. For more information on generating a certificate/private key and configuring SAP HANA, see the Tableau Community.

**Note:** The SAP HANA driver version 1.00.9 or later must be installed on Tableau Server in order to use SSO for SAP HANA. The driver cannot encrypt the SAML assertion, so you may want to enable encryption for the SAML connections. For more information, see the Tableau Community.

**Configure SSO for SAP HANA**

To configure Tableau Server to use SSO for SAP HANA:

1. Place certificate files in a folder named SAML, parallel to the Tableau Server 10.5 folder. For example:

   C:\Program Files\Tableau\Tableau Server\SAML

   You should use this location because the user account that runs Tableau Server has the necessary permissions for accessing this folder.

2. After you install Tableau Server, run the Configuration utility (Start > All Programs > Tableau Server 10.5 > Configure Tableau Server), and then click the SAP HANA tab.

3. Select **Use SAML to enable single sign-on for SAP HANA** and provide the location for each of the following:

   **SAML certificate file**—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.

   **SAML private key file**—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.
4. Select the format of the user name.

5. Select the case for the user name. This determines the case of the name when it is forwarded to the SAP HANA identity provider (IdP).

**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. In other words, if Jane Smith’s Tableau Server user account has a username of MyCo\jsmith and Tableau Server is using Active Directory for user authentication, her username on the SQL Server database must also be MyCo\jsmith and 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). This SQL Server account must also be one of two accounts on the Tableau side of things:

- The Tableau Server Run As User account (see Impersonate with a Run As User Account).
- 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 User Account**

Impersonating via a Run As User account is the recommended way to perform impersonation. The Run As User account is an Active Directory (AD) account the Tableau Server service can run under on the machine hosting Tableau Server (see Run As User). 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 account for impersonation gives the administrator the most control.

To set up impersonation with a Run As User account:

1. When you configure Tableau Server as part of Setup, under **Server Run As User**, enter the Run As User AD account that has IMPERSONATE permission for the user accounts. Under **User Authentication**, select **Use Active Directory**:
2. Click OK to finish configuration.

3. 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:

4. In Tableau Desktop, publish the workbook to Tableau Server (Server > Publish)
Workbook).

5. 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:

6. Click **OK**.

7. 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:

![Embedded Credentials](image)

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:

   ![Authentication Dialog Box]

   - **Data Source**  | **Authentication**  | **Extract**  | **Username**
   - TestV1          | Prompt user         | sqUser       |
   - TestV2          | Embedded password   | sqUser       |
   - TestV3          | Impersonate via embedded password |

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.

**Manage Content Access**
Set a Site’s Web Authoring Access

Tableau Server administrators can specify at the site level whether to allow users to edit published views in the web environment.

By default web authoring functionality is enabled for all sites. Users with the Web Edit capability to 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 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.

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.

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 interchangeably.

**Why allow users to work on the server 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 server, 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 server 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 server, many Tableau administrators use projects to allow varying levels of access to content. For example, one project can 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 Publisher site role allows both Save (overwrite) and Save As/Download.

Publishers can save (overwrite) only workbooks that they own. If a publisher edits a workbook owned by another user, only the Save As command is available, and they can save the workbook with a new name.

- The Interactor site role allows downloading but not saving (neither overwriting existing nor saving changes to a new workbook).

Capabilities set on content

These capabilities coordinate with the site role to allow levels of editing and saving.

- **Web Edit** determines whether users can edit workbooks directly on the server.

- **Download Workbook/Save As** determines whether users have access to the Save As option while they are editing a workbook, or can download a workbook as a Tableau packaged workbook (.twbx).

  Save As allows users to save changes they've made to an existing workbook as a new workbook on the server. It also determines whether they can open a workbook on the server from Tableau Desktop.

  Tableau treats saving like publishing, so to save new workbooks, users must have a site role of Publisher.

- **Save** determines whether users can save changes to an existing workbook on the...
To overwrite a workbook with changes made on the server, a user must have a site role of Publisher.

Configure site roles and permissions

Use the tables in the following sections to coordinate the site role with saving and editing capabilities.

Set these capabilities at the project level, using permissions rules.

Allow publishers to edit, save changes to, and download existing and new workbooks.

Apply these capabilities to users who have the Publisher site role.

<table>
<thead>
<tr>
<th>Capability</th>
<th>For the project</th>
<th>For specified workbooks in the project</th>
</tr>
</thead>
<tbody>
<tr>
<td>Web Edit</td>
<td>-</td>
<td>Allowed</td>
</tr>
<tr>
<td>Download Workbook/Save As</td>
<td>-</td>
<td>Allowed</td>
</tr>
<tr>
<td>Save</td>
<td>Allowed</td>
<td>Allowed</td>
</tr>
</tbody>
</table>

Allow publishers to edit, save changes to, and download new workbooks, but not overwrite existing workbooks.

Apply these capabilities to users who have the Publisher site role.

<table>
<thead>
<tr>
<th>Capability</th>
<th>For the project</th>
<th>For specific workbooks in the project</th>
</tr>
</thead>
<tbody>
<tr>
<td>Web Edit</td>
<td>-</td>
<td>Allowed</td>
</tr>
<tr>
<td>Download Workbook/Save As</td>
<td>-</td>
<td>Allowed</td>
</tr>
<tr>
<td>Capability</td>
<td>For the project</td>
<td>For specific workbooks in the project</td>
</tr>
<tr>
<td>----------------------</td>
<td>-----------------</td>
<td>---------------------------------------</td>
</tr>
<tr>
<td>Web Edit</td>
<td>-</td>
<td>Allowed</td>
</tr>
<tr>
<td>Download Workbook/Save As</td>
<td>-</td>
<td>Allowed</td>
</tr>
<tr>
<td>Save</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

The **Save** and **Save As** capabilities will be denied by the Interactor site role, even if you allow them on the content.

**See also**

- Set Permissions on Individual Content Resources
- Set Users’ Site Roles
- Permissions Reference
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.
Start by creating a project team and adopting a permissions strategy

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:

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. The Default project works as a template for new projects in the site. 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. In the menu at the top of the page, select Projects.

2. Open the permissions for the Default project. Open the Actions menu (...), and then
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.** 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 publishers whose job it is to approve or certify data models or reports.

- **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 site role have access to everything on the site, regardless of the groups you add them to.

If you have multiple 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 columns.

      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 leaders 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 leaders 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 instead of Publisher.

- 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.

---

**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 on the project thumbnail, as well as the project-specific Details tab.
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 tedious. To automate these processes, as well as make them repeatable for future updates, you can perform these tasks using REST API or tabcmd commands.

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 Tuning & Operations.
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 other Tableau customers address governance and self-service

The following list contains links to data governance presentations that Tableau customers gave at the Tableau Conference in Austin, TX, November 2016.

Server Admins: Don’t Fear Web Authoring—Robb Yeager of Sprint

Building a COE at Wells Fargo, Next Level Governance at Large Scale—Angie Greenhaw and Brian Mooneyham of Wells Fargo

Maintaining Governance in a Self-Service Environment—Mark Jackson of Piedmont Healthcare

To get access to these and related free talks, you complete a simple registration form on the TCLive website.

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.
In this article

Why use projects
Project-level administration
How the Default project acts as a permissions template
Prevent publishers from changing permissions on content they own
Next steps

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.

- 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, so you 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 an administrator or project owner. You can assign Project Leader permissions to users with a site role of Interactor or Publisher.
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

In addition to clarification points in the following list, server or site administrators can perform all of these actions.

• 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.

• 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 a project or project hierarchy.

- Run, add, or remove extract refresh schedules.

See also Permissions in Project Hierarchies

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 the project’s workbooks and data sources.

**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.

**Next steps**

- Add Projects and Move Content Into Them
- Permissions in Project Hierarchies
- Set Project Default Permissions and Lock the Project

**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 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. 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 Permissions in Project Hierarchies.

In this article

Add a top-level or child (nested) project
Move a workbook or data source to another project
Move a project or project hierarchy
Delete a project
Required access level for moving content

Add a top-level or child (nested) project

1. While you’re signed in to Tableau Server as an administrator or project leader, select Content > Projects.

2. In the Projects page, do one of the following:
   • (Administrators only) To add a top-level project, click New Project.
   • To add a child project under a top-level project, select the top-level project, and on its Projects tab, click New Project.

3. Enter a name and description for the project, and then click Create.
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, navigate to Details, and then click Edit Description.

Move a workbook or data source to another project

The following procedure is specific to workbooks; however, you follow similar steps for data sources on the Data Sources page.
1. Go to the **Workbooks** place, select one or more workbooks, and then select **Actions** > **Move**.

2. Select the new project for the workbook, and then click **Move**.

**Move a project or project hierarchy**

1. In the thumbnail or list view, navigate to the project you want to move, select the **Actions** button, and then select **Move**.
2. Select the destination project and then click **Move Project**.
How moving projects affects 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. Click **Content > Projects**. On the **Projects** page, do the following:
   a. Select the check box next to a top-level project.

   Or navigate to a child project, and on its **Projects** page, select the check box for the child project you want to delete.

   b. Select **Actions > Delete**.
2. 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.
Required site role

To move content, users must have one of the following site roles:

- Server Administrator (Tableau Server only)
- Site Administrator
- Publisher

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

Users with the Publisher site role need to 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

Users with Publisher site role 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.
Set Project Default Permissions and Lock the Project

As an administrator or project leader, you can organize a related workbooks and data sources in projects. You can then set a project’s permissions—including the default permissions that will be set on child projects you create under that project, and workbooks and data sources published anywhere in that project hierarchy. You can further lock the top-level project, at which point only administrators and project leaders can change permissions, and only at the top-level project.

Administrators and users with the Project Leader permission can lock permissions to a project. For more information, see Permissions in Project Hierarchies.

In this article

- Set the default permissions for a project
- Lock or unlock permissions on a project
- Permissions roles you can set at the project level
- Capabilities you can set for a project

Prerequisite

The remaining sections in this article assume that you have already created your user groups. For a best practice recommendation for creating groups, see Steps to coordinate projects and groups in the topic Configure Projects for Managed Self-Service.

For related information, see the topics listed at the end of this article.

Set the default permissions for a project

These steps focus on our recommended practice of assigning permissions to groups; however, you can assign permissions to individual users the same way.

Before you can complete these steps, you must create your groups, as mentioned in Prerequisite.
1. While you’re signed in to Tableau Server as an administrator or project leader, select **Content > Projects**.

2. In the **Projects** page, on the actions menu (…) for a project, select **Permissions**.

3. Click **Add a user or group rule**, select **Group**, and then select the group or name from the list.

   ![User Group Table](image)

   For an existing group, open the actions menu, and then select **Edit**.
4. Select a permissions role for Project, Workbooks, or Data Sources, and then click Save.

To create a custom set of capabilities, select the Project, Workbooks, or Data Sources labels to expand the permissions view. Click capabilities to set them to Allowed, Denied, or Unspecified. Click Save.

The following image shows how to set project permissions. The same general steps apply for workbooks and data sources.
Note: To change the settings after saving, open the group’s or user’s actions menu (…), and then select Edit.

5. View the effective permissions:

Select a group name or user name in the permission rules to see the resulting user permissions.

Expand the Project, Workbooks, or Data Sources permissions views to see individual capabilities.
Hover over a capability box to see a tooltip with details on whether a capability is allowed or denied.

6. Follow the same steps to configure additional permission rules for more groups.

Lock or unlock permissions on a project

1. Sign in to your Tableau Server site as an administrator or project leader, in the Projects area, click a project’s Actions (…) menu, and then select Permissions.
2. Click **Edit Content Permissions**.

3. In the **Content Permissions in Project** dialog box, select **Locked to the project**, and then click **Save**.
4. To unlock content permissions for the projects, click **Edit Content Permissions** again, and select **Managed by owner**.
The default permissions are reapplied to workbooks and data sources in the project, and their permissions are now editable.

Permissions roles you can set at the project level

Permissions roles are built-in templates that assign a common set of capabilities on content. You can assign them only at the project level, for the project itself, and the workbook and data sources in it.

<table>
<thead>
<tr>
<th>Template</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project Leader</td>
<td>Allows the user or group full access to the project, its child projects, and content published into that project hierarchy.</td>
</tr>
<tr>
<td>Editor</td>
<td>Allows the user or group to connect to, edit, download, delete, and set permissions for data sources or workbooks in the project. They can also publish data sources, and as long as they are the owner of a data source they publish, can update connection</td>
</tr>
</tbody>
</table>
information and extract refresh schedules. This permission is relevant for views when the view they access connects to a data source.

<table>
<thead>
<tr>
<th>Capability</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Publisher</td>
<td>Allows the user or group to publish workbooks and data sources to the project.</td>
</tr>
<tr>
<td>Connector</td>
<td>Allows the user or group to connect to data sources in the project.</td>
</tr>
<tr>
<td>Viewer</td>
<td>Allows the user or group to view the workbooks and views in the project.</td>
</tr>
<tr>
<td>None</td>
<td>Sets all capabilities for the permission rule to <strong>Unspecified</strong>.</td>
</tr>
<tr>
<td>Denied</td>
<td>Sets all capabilities for the permission rule to <strong>Denied</strong>.</td>
</tr>
</tbody>
</table>

Capabilities you can set for a project

The three capabilities you can set specifically for a project are: **View**, **Save**, and **Project Leader**.

<table>
<thead>
<tr>
<th>Capability</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>View</td>
<td>Allows the user or group to view the workbooks and views in the project. The View capability must also be allowed for the individual workbooks and views in the project.</td>
</tr>
<tr>
<td>Save</td>
<td>Allows the user or group to publish workbooks and data sources to the server and overwrite content on the server. The Save capability must also be allowed for the individual workbooks and data sources in the project. When allowed, the user with a site role that supports publishing can re-publish a workbook or data source from Tableau.</td>
</tr>
</tbody>
</table>
Desktop, thereby becoming the owner and gaining all permissions.

Subsequently, the original owner’s access to the workbook is determined by that user’s group permissions and any further permissions the new owner might set.

This permission also determines the user’s or group’s ability to overwrite a workbook after editing it on the server. For related information, see Set Web Edit, Save, and Download Access on Content.

| Project Leader | Allows the user or group to set permissions for all items in the project, lock project permissions, and edit default permissions. |

See also

Content Permissions and Ownership

Use Projects to Manage Content Access

Configure Projects, Groups, and Permissions for Managed Self-Service

Permissions in Project Hierarchies

As an administrator or project leader, you can decide how permissions are set in a project hierarchy you manage. On the top-level project in a hierarchy, you can set the project to one of two available states: locked and managed by owner. Locking permissions can help you maintain a predictable permissions model within a project hierarchy. For example, it denies publishers the option to set explicit permissions on workbooks and data sources they own.

In this article

Project permissions states: locked and managed by owner
How permissions are evaluated in project hierarchies

Project permissions states: locked and managed by owner

You can lock permissions in project hierarchies at the top level of the hierarchy. When you lock permissions, the top-level project’s permissions settings are applied to all workbooks, views, and data sources in the hierarchy.

In a locked project hierarchy, only administrators and project leaders can modify permissions, and they can do so only on the top level. Publishers cannot set permissions on their content during or after publishing from Tableau Desktop.

When permissions are *managed by the owner* ("unlocked"), they become editable by content owners and other users with the appropriate access level. During publishing, users have the option to set permissions explicitly on the workbook or data source they are publishing, unless you explicitly deny this capability on groups they are members of or individual users.

To clarify, owners always get full read, write, and other access to the content they publish, but in a locked state, they cannot change permissions on it.

**Note:** If a workbook, data source, or project with editable permissions is moved to a locked project, the permissions set on the locked project are applied to that content. The moved content’s permissions are then locked.

When to use each state

In general permissions are easier to manage when you lock projects. You need to go only to one place to see how they’re set for all content in that project’s hierarchy.

However, this requires that all child projects have the same permissions as set at the top level. The following common scenarios are possible only when you keep the project hierarchy *unlocked*:
• You want to create projects for different types of access your team needs for content in child projects. This strategy is described in Configure Projects, Groups, and Permissions for Managed Self-Service, and it requires setting unique permissions on each child project.

If you keep permissions unlocked, you can still set permissions at a parent project level that you want all child projects to take by default. And you can use groups to set capabilities explicitly, including denying the Set Permissions capability, so that users cannot set permissions on content they own.

• In web editing, you want to allow publishers to save changes to workbooks but not overwrite them.

For more information, see Allow publishers to edit, save changes to, and download new workbooks, but not overwrite existing workbooks.

How permissions are evaluated in project hierarchies

The following diagram shows how ownership and Project Leader permissions are applied in a hierarchy.

- PL = Project Leader
- O = Owner
- Dotted line indicates implicit permissions “inherited” at a parent level.
- * = Owner was assigned explicitly by the owner of the parent project.
How permissions are applied in locked project hierarchies

You can lock permission only at the top-level in a project hierarchy, and the following behavior applies:

- All child projects you create within a top-level project use the default permissions set at that top-level project (not the Default project), and you cannot modify permissions at the child-project level.

- Workbooks, views, and data sources also use the default permissions set on the top-level project.

- Users, including content owners, cannot edit permissions for individual workbooks, views, or data sources.

When users publish workbooks and data sources from Tableau Desktop, they cannot set permissions in the Publish dialog box, and their content gets the default permissions for the project they publish it to.

- Users or groups that are assigned the Project Leader permissions role at the top-level
project get Project Leader access to all child projects and their workbooks and data sources.

- Administrators and content owners can change ownership at any level in the hierarchy.

- Administrators and project leaders can edit permissions at the top-level project, and those changes propagate to all child projects.

How permissions are applied in unlocked project hierarchies

If the top-level project’s permissions are not locked, you can set permissions on its child projects in the same way you can on top-level projects, with the exception of locking the child project.

When you create a child project, it takes default permissions from the parent project. Users implicitly get the same permissions they have on the parent project.

If you set permissions at the child level, those settings take precedence over permissions set at the parent level.

See also

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)

Add a Project Image

Projects can have images that are displayed in thumbnail view in Tableau Server.
Set a project image

1. Sign in to a site on Tableau Server. On the **Content** tab, select **Projects**, and then select a project.

2. Navigate to **Details**, and then click **Edit Description**.
3. In the **About** field, 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.

4. Click **Save**.
Content Permissions and Ownership

In Tableau Server, you set content permissions to specify who is allowed to work with which content resources on a particular site.

For example, you can tightly restrict who has access to your company’s financial information, but widely share organizational development content.

Content resources on which you will generally assign permissions include the following:

- Projects
- Workbooks
- Data sources

Views in a workbook inherit permissions from the workbook, although you might have an edge case for which you want to set permissions for a single view.
What makes up a user's permissions

When Tableau determines which tasks (or *capabilities*) a user is allowed to perform on a content resource, it takes the following pieces into account:

**Site roles**: At the time that you add users to a site, you must apply a site role to them. This is the only setting that you apply to users (as opposed to content) to affect permissions. The site role determines whether the user can publish, interact with, or only view published content on that site. See Set Users' Site Roles.

**Permission rules and templates**: You assign content access through permission rules. These rules describe the capabilities that you want a user or group to be able to perform on a set of content. Examples of capabilities include editing a view or connecting to a data source. Tableau provides a set of templates for common permission roles, such as Editor, Project Leader, and so on.

**Content ownership**: By default, the person who publishes a data source or workbook to the server is the owner of that content. Ownership changes when another publisher updates the content on the server, or republishes it from Tableau Desktop. An administrator or project leader can change ownership or set defaults for the project.

**User permissions**: are the *effective* permissions that are the result of evaluating rules and settings, and which ultimately determine what a user can do with the content.

Who can set permissions

Users can have the **Set Permissions** capability. By default, these are:

- Administrators, for all content on the site.
- Project leaders, for all content in the project and its child projects.
- Content owners, generally those who create projects or publish data sources or workbooks.
Quick Start: Permissions

You can use permission rules to control access to content on a site. A permission rule is a set of 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 create permission rules for groups, and assign the permissions 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 Select a project

On the Content page for a site, navigate to the Projects page. On the Actions menu, select Permissions to view the permission rules for the project. By default, only an All Users group is defined, with pre-defined permissions roles applied to the group.

The image below shows an All Users group that was modified to remove those pre-defined permissions, in preparation for creating new groups for your own environment.

3 Create a Permission Rule

Click Add a user or group rule, select Group, and then find and select the group.

Select a permission 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 User Permissions

After you save the permission 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.

Site roles

A user's site role determines the maximum permissions allowed for that user.

- Server and site administrators can access all site content with full permissions.

- Owners always get full access to the content they've published, but can change permissions for their workbooks and data sources only when the parent project permissions are not locked.

For more information, see Set Users' Site Roles.

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
- Projects and Content 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, 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.

In this article
Considerations for changing content ownership
Who can change or be given ownership, by content type
Change a workbook or data source owner
Change a project owner

Considerations for changing content ownership

- If you want to remove a user from Tableau Server, make sure they do not own any content assets before you delete them.

  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 they continue to show as the content owner.

- If you change the ownership of a workbook or data source that includes embedded credentials (for connecting 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 workbook or data source, open the item in Tableau Desktop...
to update the embedded credentials, and then re-publish the workbook or data source.

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.

<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><strong>Top-level projects</strong></td>
<td>Server administrator¹</td>
<td>Server administrator</td>
</tr>
<tr>
<td></td>
<td>Site administrator</td>
<td>Site administrator</td>
</tr>
<tr>
<td></td>
<td>Project leader or owner</td>
<td>Publisher</td>
</tr>
<tr>
<td><strong>Child (nested) projects</strong></td>
<td>Server administrator</td>
<td>Any administrator or user of the site, excluding Guest users.</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><strong>Workbooks and data sources</strong></td>
<td>Server administrator</td>
<td>Any administrator or user of the site, excluding Guest users.</td>
</tr>
<tr>
<td></td>
<td>Site administrator</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Workbook or data source owner</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Project leader or owner of the project that contains the item</td>
<td></td>
</tr>
</tbody>
</table>

¹ The Server Administrator site role applies to Tableau Server only; not Tableau Online.
Change a workbook or data source owner

1. On the Content page for a site, select **Workbooks** or **Data Sources**.

2. Select one or more workbooks or data sources, and then select **Actions > Change Owner**.

3. Type the name of a user or select a user from the list.
4. Click **Change Owner**.

Change a project owner

1. On the Content page for a site, select **Projects**.

2. Select one or more projects, and then select **Actions > Change Owner**.
3. Type the name of a user or select a user from the list.
4. Click **Change Owner**.

**View or Edit Permission Rules and User Permissions**

When you specify permissions for a project, workbook, view, or data source, you use a permission rule to express who is allowed to work with that resource. Permission rules are the explicit capabilities that can be set for an individual user, or for a group—for each resource.

You work with these rules in the **Permissions** window, described in the steps below. You define permission rules in the upper section. The **User Permissions** section below it shows the effective, or resulting, permissions after Tableau evaluates your rule.

**View or edit a permission rule for a selected content resource**

1. On the Content page for a site, select **Projects, Workbooks, Views**, or **Data Sources** to display all items of that content type.

2. Select the check box for the item, and then select **Actions > Permissions**.
This shows the current permission rules.

3. To edit a rule, do the following:
   a. Select the actions menu (…) next to the rule name, and then select **Edit**.
   b. Click a capability in the rule to set it to **Allowed** or **Denied**, or **Unspecified**.
      See How the permission rules settings work together below for more information.
   c. Click **Save**.

4. To see the resulting permissions for the selected group or user, do the following:
   a. In the permission rules area, select the group or user name.
   b. In the **User Permissions** area, hover over a capability box to show a tooltip that indicates the capability name, its resulting setting, and how the result was determined.
The following image shows permissions on a selected workbook for Andrew Allen in the Finance group. The tooltip indicates that a rule assigned to a group he is a member of denies the web edit capability.

How the permission rules settings work together

The sections shown in the Permissions Rules area depend on the type of content you select. The settings described here are those that appear when you select a workbook or view. (Some of these settings are not available until you open a rule for editing, as described in the steps above.) To learn how to set permission rules at the project level, see Set Project Default Permissions and Lock the Project.

- **User / Group:** Lists users or groups that a rule applies to. If the permissions are not locked at the project level, you can click **Add a user or group rule** to configure permissions for those users on the selected content.

- **Permissions:** Lists available permission-role templates for the selected content element. Each template contains a predefined set of capabilities for the rule. If the capabilities selected for the user or group do not match a predefined template, the template name changes to **Custom**. The word Custom appears regardless of how the capabilities are modified.

- **View / Interact / Edit:** These are categories for the groups of capabilities that you
can set to **Allowed, Denied, or Unspecified**. Setting **Unspecified** evaluates to **Denied** if no other permissions are specified for a user or group on the content.

Although some of the names here are the same as or similar to site or permission role names, the categories here are independent of those things, and they only indicate groups of capabilities that are typically configured as a set.

**User Permissions**

Effective user permissions for a content resource are determined by the following:

- The maximum capabilities allowed for a user’s site role. The site role acts as the "ceiling" for what permissions are allowed. For more information, see Set Users’ Site Roles.

- Whether the user owns the content item

- The evaluation of each user or group permission rule that applies to that user for that content item

For example, if a user is granted Editor-level permissions for a workbook (which allows all available capabilities), but has the site role of Viewer and does not own the workbook, the user will only be allowed the capabilities of **View, Export Image, Summary Data, View Comments, Add Comments, and Save**.

In the following example, a permission rule has been created for the Finance group. The permission role template of **Editor** was initially applied to the group, which granted all capabilities. The administrator then set the **Save** capability to **Denied**, so the name for the set of permissions applied to the group became **Custom**. The **User Permissions** section for the Finance group shows that most of the users in the group have all capabilities, except for the **Save** capability. One user has even fewer capabilities because that user has a site role of Viewer.
Note that the All Users group permission rule in this example has been set to None, which leaves all of the permissions as Unspecified for the All Users group. This approach requires the administrator to specifically assign permissions for only the groups or users that should see the content.

Set Permissions on Individual Content Resources

As an administrator, or as a user granted the Set Permissions capability on a specific resource, you can change permissions on that resource (data source or workbook).

Generally we recommend setting permissions at the project level for all content in the project. A reason you might want to deviate from this practice, for example, is to explicitly deny the some capabilities on a data source or workbook that has a data source or user filter that enables a form of row-level security.
In this article
Capabilities you can set on data sources
Capabilities you can set on workbooks and views
Set permissions on a content resource
Set Permissions on Individual Content Resources

Capabilities you can set on data sources

Use permission rules to set the following capabilities for a data source:

<table>
<thead>
<tr>
<th>Capability</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>View</td>
<td>View the data source on the server.</td>
</tr>
<tr>
<td>Connect</td>
<td>Connect to the data source. The <strong>Connect</strong> permission allows a user to connect to a data source from an editor (in Tableau Desktop or Tableau Server web editing).</td>
</tr>
<tr>
<td></td>
<td><strong>Note:</strong> If a workbook author embeds credentials in a workbook or view, users who also have the <strong>Web Edit</strong> permission will be able to access to the workbook’s data source regardless of their <strong>Connect</strong> permissions.</td>
</tr>
<tr>
<td>Save</td>
<td>Publish data sources to the server and overwrite data sources on the server.</td>
</tr>
<tr>
<td>Download Data Source</td>
<td>Download the data source from the server.</td>
</tr>
</tbody>
</table>

**Note:** 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. You must explicitly grant the **Download** permissions regardless of the permissions role you apply. For more information, see Cube Data Sources.

<table>
<thead>
<tr>
<th>Delete</th>
<th>Delete the data source.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Set Permissions</td>
<td>Grant or deny permissions for the data source.</td>
</tr>
</tbody>
</table>

**Permission role templates for data sources**

<table>
<thead>
<tr>
<th>Template</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connector</td>
<td>Allows the user or group to connect to the data source on the server.</td>
</tr>
<tr>
<td>Editor</td>
<td>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.)</td>
</tr>
<tr>
<td>None</td>
<td>Sets all capabilities for the permission rule to <strong>Unspecified</strong>.</td>
</tr>
<tr>
<td>Denied</td>
<td>Sets all capabilities for the permission rule to <strong>Denied</strong>.</td>
</tr>
</tbody>
</table>
Capabilities you can set on workbooks and views

The list of capabilities and the available permission role templates vary depending on whether you are setting permissions for a workbook or a view. For information about capability definitions, see Permissions Reference.

Editing view-level permissions

When a Tableau Desktop author publishes a workbook with the **Show Sheets as Tabs** enabled, these tabbed views take on the workbook permissions rules. Changes you make to the workbook permissions affect all of its tabbed views.

To edit an individual view’s permissions, save the workbook again without tabs (or hide sheets). The default permissions are applied to the workbook, and you can then edit view permissions.

We recommend that you set view-level permissions sparingly, as an exception. Try to manage permissions at the project level as much as possible. When permissions are locked to a project, views in a workbook use the workbook permissions.

Permission role templates for workbooks and views

<table>
<thead>
<tr>
<th>Template</th>
<th>Applies to...</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Viewer</td>
<td>workbooks, views</td>
<td>Allows the user or group to view the workbook or view on the server.</td>
</tr>
<tr>
<td>Interactor</td>
<td>workbooks, views</td>
<td>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.</td>
</tr>
<tr>
<td>Editor</td>
<td>workbooks, views</td>
<td>Sets all capabilities for the rule to <strong>Allowed</strong>.</td>
</tr>
</tbody>
</table>
### Set permissions on a content resource

1. In the Data Sources, Workbooks, or Views page, select the check boxes for each resource, and then select **Actions > Permissions**.

The following image shows how this looks on the Data Sources page.

![Data Sources page](image)

**Note:** If you select multiple items and some of the items are read-only, you cannot view the permissions. Instead, select one item at a time.

2. Click **Add a user or group rule**, select **Group** or **User**, and then select the group or user name from the list.
3. Select a permission role template to apply an initial set of capabilities for the group or user, and then click **Save**.

4. 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.
5. Configure any additional rules you want for other users or groups.

6. View the resulting permissions.

Click a group name or user name in the permission rules to see the resulting permissions. Hover over a capability box to see a tooltip that shows whether a capability is allowed or denied, and what determined that result.
See also

How data access is evaluated for workbooks that connect to Tableau data sources

How Permissions are Evaluated

Permissions in Tableau Server are assigned to content resources—projects, workbooks, data sources, and sometimes to individual views. You use permission rules to specify who can work with a content resource.

What users can access and the actions available for each content type, are determined by the following:

- **Site role.** A user's site role determines whether a user can publish, interact with, or only view resources. For more information, see Set Users' Site Roles.

- **Content permissions.** Every project, workbook, data source, or view can have a unique set of permission rules.

  A permission rule includes the user or group, and the set of capabilities you want to grant users for a resource (such as the ability to edit a view). Each permission role template (such as Editor, Interactor, Viewer) specifies a predefined set of capabilities for the rule. If the capabilities that are selected do not match a predefined template, the permission role template changes to Custom.

  Available capabilities vary depending on the resource. Capabilities can be set to Allowed, Denied, or Unspecified. Denied always takes precedence over Allowed, and Unspecified results in Denied if no other permission rules allow a capability for a user.

- **Ownership.** Content owners always get full access to the content they've published. In projects with locked permissions, content owners cannot edit permissions for their workbooks and data sources.
Users with the **Set Permissions** capability can change permissions for content items in projects that aren’t locked. Administrators, content owners, and users with the **Project Leader** capability automatically have the **Set Permissions** capability.

You can set permission rules for an individual user or group for each resource. This diagram illustrates how permission rules are evaluated in Tableau Server.

A user’s effective permissions for a given content resource are determined by the following:

- The maximum capabilities allowed through the site role.
- Whether the user owns the content item.
- The result after Tableau evaluates permission rules applied to that user and all groups the user is a member of.

Notes on permissions

- Server and site administrators can access all the resources in a site with full permissions.
- You cannot set permissions at the site level; permissions are assigned to resources only.
- As content owners, publishers get full access to their content. Administrators can manage permissions on content after it’s published.

Another option for practicing content governance is to lock permissions at the project level. This does not change publishers’ ownership of the content they publish after the
project is locked. However, it enables default permissions for the users and prevents publishers from being able to change those default permissions during the publishing process. See Permissions in Project Hierarchies.

- Individual user permissions on resources take precedence over group permissions on resources. In other words, user permissions trump group permissions.

- Workbook permissions serve as templates for view permissions. When content permissions are locked to the project, and when a workbook uses tabbed views, views inherit their workbook permissions. When permissions are not locked, and when a workbook is saved without tabs, the workbook and view permissions can be edited independently.

- Project default permissions serve as templates for content in a project. When content permissions are locked to the project, the workbooks and data sources always use the default permissions. When permissions are not locked, workbook and data source permissions can be edited independently.

- For each content item, every site user is automatically included in the **All Users** group. As a result, the All Users permission rule affects how permissions are evaluated for users when you create additional group permission rules for that content item.

If you use Tableau Server in an environment where openly sharing knowledge and information across the organization is important, set the permission rule for the **All Users** group in the **Default** project to the **Publisher** permission template. Users can publish to and consume content from new projects.

If you use Tableau Server in an environment where restricting access is important, set the permission rule for the **All Users** group in the **Default** project to the role of **None**. Then, add explicit permissions for groups and users to allow them to publish and work with content in new projects.
The order of precedence in which Tableau evaluates permissions

1. **Server and Site Administrator**: Administrators can access all site content with full permissions.

2. **User - Unlicensed, Viewer license, or Guest**: If a user is Unlicensed, has a Viewer license (different than Viewer site role), or is a Guest, there are certain capabilities they are never allowed to perform. If the capability is explicitly denied for the user because of licensing, they are denied.

3. **Project Owner**: If the user owns the project that contains the content, the capability is allowed. Otherwise,

4. **Project Leader**: If the user has the Project Leader capability, or is in a group that has the Project Leader capability, they are allowed. If the user is explicitly denied the Project Leader capability, they are denied. Otherwise,

5. **User - Authorizable Owner**: If the user is the owner of the content, they are allowed. Otherwise,

6. **User - Capability Denied**: If the user has been explicitly denied the capability for the content, they are denied. Otherwise,

7. **User - Capability Allowed**: If the user has been explicitly allowed the capability for the content, they are allowed. Otherwise,

8. **Group - Capability Denied**: If the user belongs to a group that has been explicitly denied the capability for the content, they are denied. Otherwise,

9. **Group - Capability Allowed**: If the user belongs to a group that has been explicitly allowed the capability for the content, they are allowed. Otherwise,

10. The user is denied access to the content.
Permissions Reference

Administrators and other authorized users can allow or deny capabilities on content published to Tableau Server. When permissions are not locked to a project, Tableau Desktop users who publish to that project can set permissions on their content during the publishing process.

Site and permissions roles that allow full content access

Capabilities available for each content type

How data access is evaluated for workbooks that connect to Tableau data sources

Site and permissions roles that allow full content access

The following list describes roughly who gets full access to content on the server and the scope of that access:

- Server administrators (Tableau Server only) are allowed all capabilities on all content published to the server.
- Site administrators are allowed all capabilities on content published to the site.
- Groups to which you apply the Project Leader role are allowed all capabilities on content in the project.
- Publishers become the owners of content they publish, and they are allowed all capabilities on that content, with the exception of setting permissions in locked projects.

Capabilities available for each content type

The following table shows how permissions apply to content and describes each capability.
<table>
<thead>
<tr>
<th>Permission</th>
<th>Applies to…</th>
<th>When allowed, users can…</th>
</tr>
</thead>
<tbody>
<tr>
<td>View</td>
<td>workbooks</td>
<td>Open the item on Tableau Server.</td>
</tr>
<tr>
<td></td>
<td>data sources</td>
<td><strong>Note:</strong> When a workbook is configured to show sheets as tabs, all views use the workbook permissions, even if different permissions are specified on an individual view.</td>
</tr>
<tr>
<td></td>
<td>views</td>
<td></td>
</tr>
<tr>
<td></td>
<td>projects</td>
<td></td>
</tr>
<tr>
<td>Web Edit</td>
<td>workbooks</td>
<td>Edit views in workbooks. For information, see Set Web Edit, Save, and Download Access on Content.</td>
</tr>
<tr>
<td></td>
<td>views</td>
<td></td>
</tr>
<tr>
<td>Save</td>
<td>workbooks</td>
<td>Overwrite the resource on the server.</td>
</tr>
<tr>
<td></td>
<td>data sources</td>
<td></td>
</tr>
<tr>
<td></td>
<td>views</td>
<td>When allowed, the user can re-publish a workbook or data source from Tableau Desktop, thereby becoming the owner and gaining access to all permissions. Subsequently, the original owner’s access to the workbook is determined by that user’s group permissions and by any further permissions the new owner might set.</td>
</tr>
<tr>
<td></td>
<td>projects</td>
<td>This capability also determines the user’s or group’s ability to overwrite a workbook after editing it on the server.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Special consideration for the <strong>All Users</strong> group: To help protect an owner’s content from being overwritten by another user (via publishing from Tableau Desktop or saving a web-edited workbook on the server), when a user publishes into a project where the <strong>All Users</strong> group has permissions, the <strong>Save</strong> permission for the <strong>All Users</strong> group is changed from</td>
</tr>
<tr>
<td>Permission</td>
<td>Applies to…</td>
<td>When allowed, users can…</td>
</tr>
<tr>
<td>----------------------------------</td>
<td>--------------</td>
<td>------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Download Workbook/Save As</td>
<td>workbooks</td>
<td><strong>Allowed to Unspecified</strong> by default. You can then manually modify this capability if you want to allow it.</td>
</tr>
<tr>
<td>Download Data Source</td>
<td>data sources</td>
<td>Download the data source from the server.</td>
</tr>
<tr>
<td>Delete</td>
<td>workbooks</td>
<td>Delete the content resource.</td>
</tr>
<tr>
<td></td>
<td>data sources</td>
<td></td>
</tr>
<tr>
<td></td>
<td>views</td>
<td></td>
</tr>
<tr>
<td>Filter</td>
<td>workbooks</td>
<td>Modify filters in the view, keep only filters, and exclude data.</td>
</tr>
<tr>
<td></td>
<td>views</td>
<td></td>
</tr>
<tr>
<td>Add Comments</td>
<td>workbooks</td>
<td>Add comments to views in a workbook.</td>
</tr>
<tr>
<td></td>
<td>views</td>
<td></td>
</tr>
<tr>
<td>View Comments</td>
<td>workbooks</td>
<td>View the comments associated with the views in a workbook.</td>
</tr>
<tr>
<td></td>
<td>views</td>
<td></td>
</tr>
<tr>
<td>Download Sum-</td>
<td>workbooks</td>
<td>View the aggregated data in a view, or in the user's</td>
</tr>
<tr>
<td>Permission</td>
<td>Applies to...</td>
<td>When allowed, users can...</td>
</tr>
<tr>
<td>----------------------</td>
<td>----------------</td>
<td>---------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>mary Data views</td>
<td></td>
<td>selection within the view, and download that data as a text file in .csv format.</td>
</tr>
<tr>
<td>Download Full Data</td>
<td>workbooks</td>
<td>View the underlying data behind each row in a view, as restricted by any marks the user has selected, and download the data as text file in .csv format.</td>
</tr>
<tr>
<td>Download Image/PDF</td>
<td>views</td>
<td>Download each view to PNG or PDF format.</td>
</tr>
<tr>
<td>Share Customized</td>
<td>workbooks</td>
<td>Users can save customizations they’ve made to a view using the interaction features (such as filters). They can then share their custom view for others to see. Customization in this way does not change the view’s default state or its underlying data.</td>
</tr>
<tr>
<td>Move workbooks</td>
<td></td>
<td>Move workbooks between projects.</td>
</tr>
<tr>
<td>Note: Only administrators can move data sources between projects.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Set Permissions</td>
<td>workbooks</td>
<td>Specify permissions for the resource. Views in a workbook take permissions set at the workbook level.</td>
</tr>
<tr>
<td>Connect</td>
<td>data sources</td>
<td>Connect to a published data source from Tableau</td>
</tr>
</tbody>
</table>
Permission | Applies to… | When allowed, users can…
---|---|---
| Desktop or the server’s web editing environment. See also: How data access is evaluated for workbooks that connect to Tableau data sources

| Project Leader | projects | Set permissions for all resources in a project and for the project itself. Can lock project permissions and edit default permissions. Also can change content owner, move content, and run refresh schedules.

How data access is evaluated for workbooks that connect to Tableau data sources

When a Tableau Desktop user publishes a workbook that connects to a data source published to the server (a Tableau data source), the user can set the data authentication type to **Embed password** or **Prompt users**. In this specific publishing case, data authentication refers to the workbook’s connection to the Tableau data source. This is independent of the authentication set in the published data source’s connections to the underlying data.

When a user subsequently opens the workbook on the server, the workbook-specific data authentication is evaluated along with the **Connect** capability, set in permissions rules. Whether the user can see the data in the workbook is determined as follows:

- When the workbook publisher selects the **Embedded password** option, they effectively embed their own **Connect** permissions, allowing anyone who can view the workbook to see the data it connects to. How the viewer’s **Connect** capability is set on the data source does not affect this outcome.

- When the publisher selects the **Prompt users** option, viewers who are allowed the **Connect** capability are on the data source are prompted for their credentials. Users who are denied the **Connect** capability are denied access the data.

The following table summarizes the two bullet points above.
<table>
<thead>
<tr>
<th>Workbook</th>
<th>Data source authentication</th>
<th>Connect capability</th>
<th>Resulting access to data in the workbook</th>
</tr>
</thead>
<tbody>
<tr>
<td>Embedded password</td>
<td>Embedded password or Prompt user</td>
<td>Allowed or Denied</td>
<td>Allowed through the workbook publisher’s Connect capability set on the data source</td>
</tr>
<tr>
<td>Prompt user</td>
<td>Embedded password</td>
<td>Allowed</td>
<td>Allowed through the viewer’s Connect capability</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Denied</td>
<td>Denied</td>
</tr>
<tr>
<td>Prompt user</td>
<td>Allowed</td>
<td>Viewer is prompted for database credentials</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Denied</td>
<td></td>
<td>Denied</td>
</tr>
</tbody>
</table>

Manage Data

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 Linux? 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 formatting, and so on.

• **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

• 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, go to the **Data Sources** page, select the check box next to the data source, and in the upper-left of the Data Sources page, select **Actions**.

   - **Edit and view permissions:** Permissions can specify which users or groups can connect to, modify, or download data sources. As mentioned at the beginning of this section, if this action is not available, it’s likely that the project permissions are locked.

   - **Edit connection information:** Update embedded credentials or other metadata that describes the connections to the underlying data.

   - **Create a new workbook or download:** You can start a new workbook in the browser environment by connecting to a Tableau data source. Or download the data...
source to use locally.

- **Change the data source owner**
- **View the data source’s revision history**
- **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.

- **Add or remove keyword tags**: Tags can contain a single word or multiple words, delimited by a comma.
- **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.
- **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.

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 10.5 and how the change impacts your users.
In this article

Changes to extracts in version 10.5
Tasks that cause extract upgrade
Impact of extract upgrade
What to expect after extract upgrade

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 10.5 use the .hyper format. Though Tableau version 10.5 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.
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 10.5. 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 refresh
- 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

Impact of extract upgrade

Tableau recommends that the Tableau Desktop version in your environment be upgraded to match Tableau Server 10.5. 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 Desktop 10.4</td>
<td>√</td>
<td>Not possible</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>
## 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>Download and open in Tableau Desktop 10.4</td>
<td>✓</td>
<td>Can't open workbook; you see a &quot;this workbook uses a .hyper extract and is not compatible with this version; open the workbook in version 10.5 or later&quot; error message, and then asked to locate the extract</td>
</tr>
<tr>
<td>Download and open in Tableau Desktop 10.5</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>Export As Version from Tableau Desktop 10.5</td>
<td>✓</td>
<td>✓</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>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>using 10.4 tabcmd</td>
<td></td>
<td></td>
</tr>
<tr>
<td>using 10.5 tabcmd</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>Publish</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>using 10.4 tabcmd</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>using 10.5 tabcmd</td>
<td>Not possible</td>
<td>Not possible</td>
</tr>
<tr>
<td>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>using 10.4 Tableau Command-Line Utility</td>
<td>✓</td>
<td>Not possible</td>
</tr>
<tr>
<td>using 10.5 Tableau Command-Line Utility</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>
</tbody>
</table>
Why keep an extract in .tde format?

If the version of Tableau Desktop cannot be upgraded to match Tableau Server 10.5, 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.

- Continue to perform extract refresh and append data tasks using Tableau Server 10.5, 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 10.5. For more information, see After an extract upgrade section in Tableau Help.

View Data Source Attributes

On the Data Sources page, you can filter the view to data sources or connections and see attributes for each.
View data sources by name

To filter by data source name, in the View list, select **Data Sources**.

Attributes in the data source name list include the following:

- **Icon/Name**—The icon next to the data source name indicates whether the data source is published (✓) or 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 data sources that are named similarly, or are based on the same or similar underlying data, or are published without any descriptive information about them. When this is the case, analysts who create views in Tableau Desktop or web authoring can't be confident knowing which data source is the right one to connect to.

To help your users find the data that's trusted and recommended for their type of analysis, you can **certify** the data sources that comply with your organization's data standards.

Certification complements the Recommendations Trainer feature, by offering a way to promote data sources through curation.

**In this article**

How certification helps users find trusted data
Who can certify data sources
Certify a data source

How certification helps users find trusted data

When you certify a data source, a certification badge appears next to it on the Data Sources page, and in data source lists, such as 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 certification badge. The tooltip also shows who certified the data source and the data source’s connection type.

For more information, see the Certify a data source steps below.
Create guidelines for selecting data sources 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. 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 sources

To access data source certification, a user must have either of the following permissions levels on Tableau Server:

- Site Administrator site role.
- Publisher site role with the Project Leader permissions role on the project that contains the data sources.

Certify a data source

1. Sign in to Tableau Server and select the Data Sources tab.
2. Select the name of the data source, and then select the Details tab.
3. 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.

**Keep Data Fresh**

**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.

See also

Enable Extract Refresh Scheduling and Failure Notification

**Refresh Data on a Schedule**

You can set scheduled refresh tasks for published data source extracts and published workbooks that connect to extracts.

**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.

1. When you’re signed in to Tableau Server, display **Content > Data Sources** or **Content > Workbooks**, depending on the type of content you want to refresh.

2. Select the check box for the data source or workbook you want to refresh, and then select **Actions > Extract Refresh**.

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.

By default, and if this option is not shown, a full refresh is run. Incremental refresh is available only if you configured it in Tableau Desktop before publishing the extract. For information, see Refreshing Extracts in the Tableau Help.

Click the **Schedule Refresh** button.

Note: If you want to add a new schedule, you can do so on the **Schedules** page.

**Quick Start: Manage Incremental Extracts**

When you publish a workbook that has an incremental extract, you can associate it with up to two refresh tasks that Tableau Server will handle for you: An incremental refresh of the extract and a full refresh. After you publish the workbook, you or a Tableau Server
administrator can modify any tasks that are associated with the workbook. You can also delete tasks or add more.

1 Publish and Assign a Schedule

In Tableau Desktop, after you create a workbook that uses an extract, go to Server > Publish Workbook, and click Scheduling & Authentication. Next, choose schedules for your refreshes and click OK.

After you publish in Tableau Desktop and choose your refresh schedules, Tableau Server handles the refresh tasks for you.
2 Select the Workbook

To modify a workbook’s scheduled task, sign in to Tableau Server and on the **Workbooks** page, select the workbook:

![Workbooks screenshot](image)

3 Access the Refresh Schedule

Click **Refresh Schedule**.

![Refresh Schedule screenshot](image)

Select the check box for the refresh task you want to modify:
4 Edit, Delete, or Add More Tasks

Select the action you want to take—for example, **Change Schedule**—and make your selection. You can also delete the task, change its priority, or add more refresh tasks.

**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 tabadmin set backgrounder.failure_threshold_for_run_prevention command. For more information, see tabadmin configuration options.
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):

![Alerts Menu Example](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.

  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 Publisher. If your role is Publisher, you also must be the data source owner.

1. Sign in to the site that has the data sources you want to modify, and open the Data Sources page.

2. Select the name of the data source with the connection you want to update.

   Use the search box or filters on the left to narrow the data source list. The values you type into the **Server** and **Database username** fields are treated as regular expressions.

3. In the Connections view, select the check box for the connection, and then click **Actions > 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.

For connections to Google, Salesforce, and web data connector (WDC) data, see 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/Web Save As** permission for the data source. For more information, see Set Permissions on Individual Content Resources.

- Correct drivers installed and ports opened on computer running Tableau Desktop.

Web Data Connectors in Tableau Server

Web data connectors 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. Web data connectors 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 web data connector can include internal web services, JSON data, REST APIs, and other sources that are available over HTTP or HTTPS. Users can create their own web data connectors or use connectors that were created by others.

For information about how to use a web data connector in Tableau Desktop, see [Web Data Connector](#) in the Tableau Desktop documentation.

For information about how to create a web data connector, see the [Web Data Connector documentation](#) on Github.
Before you run connectors on Tableau Server

Safe list method and import method comparison

The safe list method

The import method

Refresh the extract for a connector

Troubleshooting

Before you run connectors on Tableau Server

As a security measure, Tableau Server won't run web data connectors unless you approve the connector, as explained in this topic.

**Note:** You must be a server administrator to approve web data connectors for use on Tableau Server.

Web data connectors require your approval because they contain executable code and typically make requests to third-party websites. Before a user can use a web data connector via Tableau Server, you must either add the connectors to a safe list (to a whitelist) or import the connectors into Tableau Server. 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.

Safe list method and import method comparison

When you add a connector to the safe list (whitelist), you configure Tableau Server to allow connections to a particular URL where the connector is hosted. The connectors can then be hosted on a server inside your organization's firewall or on an external domain. The safe list is the recommended way of allowing Tableau Server to run web data connectors, and importing web data connectors is deprecated starting with version 10.5.
Alternatively, you can import a web data connector. When you import a connector, you run a `tabadmin` command that imports (copies) the connector from a location on your network to all of the machines in your Tableau Server installation.

**Note:** In versions of Tableau Server before 10.0, importing is the only way to run web data connectors on Tableau Server.

Reasons to use a safe list

You might want to add web data connectors to a safe list if:

- Your organization wants to host the connector on a separate server in your network or on an external domain. (That is, on a computer that is not running Tableau Server.)

- Your organization makes updates to connectors frequently. By adding the connector to the safe list, you avoid the need to re-import the connector each time you change it.

- The connector references many files and you do not want to import each file to Tableau Server individually.

Reasons to import web data connectors

As noted, the recommended way to configure Tableau Server to be able to run web data connectors is to use a safe list. However, you might want to import web data connectors if:

- Your organization does not have an existing web server that you can use to host the connector.

- Your organization has imported many connectors to Tableau Server in previous versions and wants to manage the connectors in a central location.

By default, both ways of configuring Tableau Server to run connectors are allowed. However, you can restrict the ways that connectors can be added or imported with the `tabadmin set webdataconnector.whitelist.mode` option. For more information, see `tabadmin set options`. 
The safe list method

To add a web data connector to the safe list, use the `tabadmin whitelist_web-dataconnector` command. This command lets you perform the following tasks:

- Add a connector to the safe list.
- List connectors on the safe list.
- Remove a connector from the safe list.
- Configure a secondary safe list, that is, a list of domains that a particular connector can send requests to and receive requests from.

For more information, see `tabadmin whitelist_webdataconnector`.

The import method

Use the `import_webdataconnector`, `list_webdataconnectors`, and `delete_webdataconnector` commands to manage imported connectors.

Import a web data connector

1. Ensure that you have the HTML file for the web data connector and any supporting files, such as `.css` files or `.js` files.

   The HTML file name for the web data connector can only contain these characters:

   `a-zA-Z0-9~.-_`

2. On the server, run the `import_webdataconnector` command, as in this example:

   ```
tabadmin import_webdataconnector connector1.html
   ```

   You can import a web data connector as a local file on the server or from a network share (for example, `\\myshare\connector1.html`), as in these examples:

   ```
tabadmin import_webdataconnector
   ```
c:\webdataconnectors\connector1.html

tabadmin import_webdataconnector \my-share\webdataconnectors\connector2.html

When the command finishes, it displays a URL, as in this example:

===== Importing web data connector to server...
    -- The web data connector with the following URL was imported to the server:
    http://myserver/webdataconnectors/connector1.html

3. Give the URL of the imported web data connector to any users who want to use that connector.

   Note: The URL displayed above uses the server's host name by default. If you have configured a DNS alias for the server, replace the server host name with the DNS alias.

Re-import a web data connector

If you want to re-import a web data connector that's already been imported (for example, you want to import an updated version of the connector), run the import_webdataconnector command with the overwrite option, as in this example:

   tabadmin import_webdataconnector \my-share\webdataconnectors\connector1.html --overwrite

The older version of the connector might still be available in the server’s cache, and users who work with the connector might still see the older version. By default, the maximum lifetime for an item in the cache is eight hours. To force a cache reset, restart the server.
List imported connectors

As the server administrator, you can see a list of web data connectors by running the following command:

```
tabadmin list_webdataconnectors
```

In order to reference a web data connector in a workbook, users need to know the URL for the connector. To get a list of connector URLs, use this command:

```
tabadmin list_webdataconnectors --urls
```

Delete imported connectors

If you no longer need a web data connector, you should delete it from the server. Use the following command to remove an individual web data connector, where `connector_name` is the name of the connector file to delete:

```
tabadmin delete_webdataconnector connector_name
```

(To see a list of web data connectors on the server, use the `tabadmin list_webdataconnectors` command).

To remove all web data connectors from the server, use the following command:

```
tabadmin delete_webdataconnector --all
```

**Note:** When you delete a web data connector, a version of the connector might still be available in the server’s cache, and users might still be able to work with the connector. By default, the maximum lifetime for an item in the cache is eight hours. To force a cache reset, restart the server.

Reference external files from imported connectors

If a web data connector .html file references external files, you must make sure that those files are available on the server. For example, a web data connector might reference an
external .css file in a <link> element or a .js file in a <script> element.

If the external files are referenced using a URL (http://), Tableau Server can access the external files as long as the files are on a server that is accessible to Tableau Server.

If the external files are referenced as local files, you can import them into Tableau Server using the import_webdataconnector command. For example, if a web data connector that you are importing references the myconnectors.css file, you import the connector and the .css file using this sequence of commands:

```
tabadmin import_webdataconnector connector1.html
```

```
tabadmin import_webdataconnector myconnectors.css
```

An important point is that all files imported using the import_webdataconnector command are stored in the same directory on the server—Tableau Server does not let you import external files into a subdirectory. Therefore, you must make sure that any local files referenced in <link> or <script> elements in the connector's .html file do not include paths, only file names.

Imported connectors in a distributed environment

If your server is configured as a cluster, web data connectors are imported to each computer where a gateway process is running. This makes the web data connector available for distributed access across your cluster. Deleting a connector in a distributed environment removes the connector from all the computers where the gateway process is running.

In a distributed environment, the process of importing or deleting a web data connector might complete only partially. If you’re importing a connector, the connector might be copied to some of the computers where the gateway process is running, but not to all of them. In that case, the tabadmin import_webdataconnector command reports the error using text like this:

```
The web data connector with the following URL has been imported to some gateways on the server, but not all.
```
Similarly, if you’re deleting a web data connector, the connector might be removed from some computers but not all of them. The `tabadmin delete_webdataconnector` command reports the error using text like this:

The web data connector was deleted from some gateways on the server, but not all.

**Note:** If the delete process is partially successful, users can still access the connector.

If the import or delete process reports partial success, you can try either of the following solutions:

- Run the import or delete process again. If you’re importing, run the `tabadmin import_webdataconnector` command again, and use the `--overwrite` option to overwrite any instances of the connector that were successfully installed. If you’re deleting, run the `tabadmin delete_webdataconnector` command again. Tableau Server will remove any remaining instances of the connector.

- Stop the server, run `tabadmin configure`, and then restart the server. The configuration process makes sure that any web data connectors are correctly distributed (imported or deleted) in all nodes where the gateway process is running. Since this option requires you to stop the server, you would choose it if it’s practical to stop the server, or if you have some other reason to stop and restart the server.

Performing site import and site export with web data connectors

Web data connectors are imported as server-wide resources; they are not associated with a specific site on your server. Therefore, if you export a site using the `tabadmin exportsite` command, the resulting .zip file does not include web data connectors that might be referenced by workbooks on the site.
Managing imported connectors for failover in a cluster

If your server is configured as a cluster with a backup primary server, you must make sure that web data connectors that you have imported to the primary are available if you need to failover to your backup primary. If the web data connectors are not available on the new primary after a failover, running the configuration process on the primary server can end up removing the connectors from other computers where a gateway process is running.

To make sure that web data connectors are available after a failover, follow these steps:

1. Make sure that you keep an up-to-date backup of the web data connectors that have been imported to your server.

2. After the primary fails, and before you start the backup primary, copy the web data connectors from the backup location to the following folder on the backup primary:

   C:\ProgramData\Tableau\Tableau Server-\data\tabsvc\httpd\htdocs\webdataconnectors

   If you have created a backup of the primary server using the tabadmin backup command, the .tsbak file created by the backup file contains the web data connectors. You can extract the contents of a .tsbak file and get the web data connectors.

   If you installed Tableau Server on a different drive, substitute that drive letter for C:.

3. Overwrite the tabsvc.yml file on the backup primary.

4. Run the tabadmin failoverprimary command. For more information, see Quick Start: Creating a Backup Primary

   If necessary, you can also reimport the web data connectors, as described earlier in this topic.

Refresh the extract for a connector

When a user creates a workbook that uses a web data connector, Tableau 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 web data connector, 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 web data connector 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.)

If you want, you can disable refresh for all web data connectors, even those that were previously imported. To disable refresh, use the `tabadmin set` command to change the `web-dataconnector.refresh.enabled` setting to `false`, as in the following example:

```
tabadmin set webdataconnector.refresh.enabled false
```

Troubleshooting

If the server experiences problems with adding connectors to the safe list or importing connectors, you can examine the `tabadmin.log` files. Be sure to check the log files on both the primary server and on the other servers that are running the gateway process. For more information about log files, see Server Log File Locations.

If the issue is that Tableau Server will not refresh an extract that was created by a web data connector, make sure that the `webdataconnector.refresh.enabled` configuration setting has been set to `true`. 
If you have re-imported a changed web data connector on the server (overwriting an existing one), but users who work with the web data connector are not seeing the changes, the users might be getting a cached version of the older version. By default, the cache is reset after eight hours; after a cache reset, older versions of the web data connector will no longer be used. If you want to force the cache to reset, you can restart the server.

If you have deleted an imported connector from the server but users are still able to work with the connector, the connector is probably still in the server’s cache. A web data connector can stay available in the cache for up to eight hours. To clear the cache, restart the server. If you delete a web data connector from a server in a distributed environment, make sure that the connector has been successfully deleted from all computers where a gateway process is running.

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
- Test the web data connector in an isolated environment
- Monitor the traffic created by the connector
- Test the performance and resource usage of the connector

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 Wireshark 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.

Navigate Site Admin Pages

When you sign in to Tableau Server as a site administrator, you see menus for site and content management that are not available to other users.

A site administrator can:

- Manage content and assign permissions.
- View and manually run schedules for extract refreshes and subscriptions.
- Manage 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.

For information about navigating content (workbook, data source, project) pages in Tableau Server, see Navigate Tableau on the Web and Manage Web Content in the Tableau Help.

Site administrator pages

Site administrators manage all site content and access on a per-site basis.
If you work with a single site on Tableau Online, or your Tableau Server deployment has one site, you’ll see menus for managing that site’s content, users, groups, schedules, and tasks, and for monitoring its status.

If you work with multiple sites, the menus across the top include a site menu on the left side.

The site menu displays the name of the current site. To go to another site, select the site menu, and then select the site name.

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

Get Started with Web Authoring

Connect to Published Data Sources While Web Editing

Build Data Views on the Web

Create a Dashboard

Create a Story

Embed Views and Dashboards in Web Pages
Manage Server

Security ................................................................................................................. 638
Preparing for Configuring OAuth Support .............................................................. 745
Configure Settings for Your Data Provider ............................................................ 746
Obtain a Client ID and Enable Google APIs ......................................................... 747
Configure Tableau Server for Google OAuth ......................................................... 748
Managing access tokens ....................................................................................... 749
Create a Connected Salesforce App ....................................................................... 750
Configure Tableau Server for Salesforce.com OAuth ................................................. 752
Managing access tokens ....................................................................................... 753
Create a Connected Intuit App ............................................................................. 753
Configure Tableau Server for QuickBooks Online .................................................. 754
Managing access tokens ....................................................................................... 755
Managing credentials centrally ............................................................................. 756
See also ................................................................................................................. 757
See also ................................................................................................................. 802
User-name mapping options ................................................................................. 804
Change the certificate mapping ............................................................................. 805
Address user-name mapping ambiguity in multi-domain organizations .............. 806
Issues ................................................................................................................... 807
We couldn't find a valid client certificate. Contact your Tableau Server ............ 807
Invalid user name or password ........................................... 808

We couldn't find your user name in the client certificate. Contact your Tableau Server administrator or sign in using your Tableau Server account. 808

Users unexpectedly see a sign-in dialog box that displays an error message 809

We couldn't find your user name in the client certificate. Contact your Tableau Server administratorCertificate does not contain a valid Tableau Server user name. ......................................................... 810

The user is signed in using an unexpected user name (LDAP mapping) …811

The user is signed in as the incorrect user (UPN or CN mapping) ……811

Manage Licenses ........................................................................ 826

Add Users to the Server ................................................................ 837

Sign in to Tableau Server Admin Pages ........................................ 841

Manage Sites .............................................................................. 846

Extract Refresh Schedules .......................................................... 882

Performance ............................................................................... 899

Monitoring Tableau Server ...................................................... 945

Maintenance .............................................................................. 978

Troubleshooting ......................................................................... 1019

Command Line Utilities ............................................................... 1104

Server Administrator Reference .................................................. 1324

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 identity store. 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 Active Directory, SAML, or OpenID. In all cases, whether authentication takes place locally or is external, each user identity must be represented in the Tableau Server identity store, which is managed by the repository.

Looking for Tableau Server on Linux? See Authentication.

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.

User identity in Tableau Server

When you install Tableau Server, you must select the process that the server will use to manage user authentication: local authentication or Active Directory. Before you install Tableau Server, you should understand how these two options impact your overall authentication strategy. After you select and set the authentication process, Tableau Server will configure the various components for the authentication method that you have selected. After this
configuration is complete, you cannot change the authentication method. In fact, to change this configuration, you must uninstall the server, delete the configuration on the computer, and then reinstall the server.

Local authentication

If the server is configured to use local authentication, then the Tableau Server identity store is used exclusively to authenticate 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”).

Use local authentication if any of the following are true:

- Your organization does not manage users with Active Directory
- You do not want to use Active Directory
- You want to use OpenID for authentication and single sign-on

When you configure Tableau Server for local authentication, you cannot set password policies or account lockout on failed password attempts. If you require these account safeguards, then you should use Active Directory authentication.

Active Directory

If Tableau Server is configured to use Active Directory authentication, then credentials are managed and verified by Active Directory. 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 local identity store.

In this scenario, Tableau Server must be installed in a domain in Active Directory. Tableau Server will sync user and group metadata from Active Directory to the identity store. You do not have to manually add users. However, after the data is synchronized, you will need to assign site and server roles. You can assign these individually, or at the group level. Tableau Server does not synchronize any data back to Active Directory. Tableau Server manages content and server access according to the site role permission data is stored in the repository.

If you are already using Active Directory to manage users in your organization, then we recommend selecting Active Directory authentication during Tableau setup to make user provisioning and management easier. For example, by synchronizing Active Directory groups, you can set minimum site role Tableau permissions for users that are synchronized in the groups. You can synchronize specific Active Directory groups, or you can synchronize them all. For more information, see Synchronize All Active Directory Groups on the Server.

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.

**Data connection 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 using local authentication, while configuring Kerberos delegation, OAuth, or SAML authentication to specific data sources. See Data Connection Authentication.

**Single sign-on options for Tableau Server**

Tableau Server supports several single sign-on (SSO) solutions. With SSO, users don't have to sign in to Tableau Server explicitly. Instead, the credentials they've used to
authenticate already (for example, by signing in to your corporate network) are used to authenticate them to Tableau Server, and they can skip the step of entering a user name and password. With SSO, the user’s identity as established externally is mapped to a user identity defined in the Tableau Server identity store.

When you configure Tableau Server to use an SSO solution, all authentication is handled by the SSO solution. However, Tableau Server will manage user access to Tableau resources based on the site roles stored in the identity store and permissions set on those resources.

Tableau Server supports these types of SSO:

- **SAML.** You can configure Tableau Server to use SAML (security assertion markup language) for SSO. 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.

  You can use SAML to access Tableau Server if you have configured Active Directory or local authentication on Tableau Server. For more information, see SAML.

- **Kerberos.** If Kerberos is enabled in your environment and if the server is configured to use Active Directory authentication, you can provide users with access to Tableau Server based on their Windows identities. You cannot use Kerberos if your Tableau Server is configured for local authentication. For more information, see Kerberos.

- **OpenID.** 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 on Tableau Server, the server must be configured to use local authentication. Active Directory authentication is not supported. For more information, see OpenID Connect.

- **Trusted Authentication.** Trusted authentication 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.

- **Integrated Windows Authentication.** If you have configured Tableau Server with Active Directory authentication, you can enable automatic logon. Automatic logon uses Microsoft SSPI to sign in your users based on their Windows user name and password. Users are not prompted for credentials, which creates an experience similar to single sign-on (SSO). To enable automatic logon see, Configure General Server Options.

Related topics

- Trusted Authentication

- REST API: Signing In and Out (Authentication)

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.

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.
In addition, if you want to use site-specific SAML, you must configure server-wide SAML before you configure individual sites.

- **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.

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:
Step 1  User navigates to the Tableau Server sign-in page or a published workbook.

Step 2  Tableau Server starts the authentication process and redirects the request to the registered IdP.

Step 3  The IdP requests the user’s username and password and authenticates the user to the IdP.

Step 4  The IdP returns a SAML success response to Tableau Server.

Step 5  Tableau Server verifies that the username in the response matches what is stored in Tableau Server and displays the page the user requested in Step 1.

Quick Start: Single Sign-On with SAML

With Tableau’s SAML support, you use one or more external identity providers (IdP) to authenticate Tableau Server users. This quick start describes how to set up a server-wide SAML implementation that uses a single IdP.

To configure Tableau Server for SAML, you need the following:
- Certificate file: A PEM-encoded x509 certificate that has a .crt filename extension.
- Certificate key file: An RSA or DSA key file that is *not* password protected and that has a .key filename extension.
- IdP account: Examples are PingFederate, SiteMinder, and OpenAM.
- Matching usernames: Tableau Server usernames and the usernames stored in the IdP must match. Ensure that the username you plan to use for your Tableau Server administrator account exists with your IdP before you run setup.

1 Specify the Server and Certificates

Run Tableau Server setup. After you configure your general settings in the Configuration utility, click the SAML tab and select **SAML authentication for the server**:

In the **Tableau Server return URL** box, enter the customer-facing URL for your installation of Tableau Server. Enter the same value for **SAML entity ID**.

Create a folder named SAML under the following folder, and then copy the .crt and .key files to the new folder.

**C:\Program Files\Tableau\Tableau Server**

Use the new folder for the **SAML certificate file** and **SAML key file** boxes.

2 Export Metadata from Tableau

Leaving the **SAML IdP metadata file** box empty, click the **Export Metadata File** button.
Use the .xml file name of your choice.

In the next dialog box, save the XML file. You will need to provide this file to your IdP in the next step.

3 Export Metadata from the IdP

On the IdP’s website, add your installation of Tableau Server as a connection type for the IdP to authenticate. As part of this, you import the Tableau metadata .xml file you created in step 2, and confirm that your IdP’s settings use `username` as the attribute element to verify.

Next, export your IdP’s metadata .xml file and copy it to the following folder on the computer where Tableau Server is installed:

C:\Program Files\Tableau\Tableau Server\SAML

4 Test the SAML Sign-On

On the SAML tab in the Tableau Configuration utility, enter the location of the IdP’s file in the SAML IdP metadata box. Click OK. Finish the setup process.
To test your changes, start a new web browser session and go to the URL for your installation of Tableau Server. If SAML is properly configured, the Sign On prompt is from your IdP and not from Tableau:

![Sign On Prompt]

Please sign on first and we'll send you right along.

Username
jsmith
Password

Sign On  Cancel

SAML Requirements

Before you configure SAML on Tableau Server, make sure your environment meets the requirements.

In this article

- Certificate and identity provider (IdP) requirements
- User management requirements
- SAML compatibility notes and requirements
- Using SAML SSO with Tableau client applications
- XML data 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, you can use the
same certificate with SAML. See About the certificate and key files later in this topic for details.

• **Certificate key file.** An RSA or DSA private key file that is not password protected, and that has the `.key` extension. This file is used by Tableau Server, not the IdP. The certificate key file must not have the passphrase. If you have an SSL certificate key file, you can use it for SAML as long as it does not have a passphrase. See About the certificate and key files later in this topic for details.

• **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. The IdP must support SAML 2.0 or later.

• **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.

### About the certificate and key files

If you are using a PEM-encoded x509 certificate file for SSL, you can use the same file for SAML. When it's used for SSL, the certificate file is used to encrypt traffic. When it's used for SAML, the certificate is used for authentication.

Tableau Server does not support certificate and certificate key files for SAML if the certificate and key require a chain file. If your SSL certificate and key files require a chain file, you need to generate new certificate and key files to use for SAML.

**User management requirements**

When you enable SAML, user authentication is performed outside of Tableau, with the IdP. However, the user management is performed either by Active Directory or by Tableau Server (called local authentication even though Tableau Server is not performing authentication in this scenario).
**Note:** The REST API and tabcmd do not support SAML single-sign (SSO). To sign in, you must specify the name and password of a user who has been created on the server. The user could have a local or Active Directory account, depending on how you have configured Tableau Server. For Tableau Online, you can specify the TableauID credentials of the user. REST API or tabcmd calls will have the permissions of the user you sign in as.

When you configure user authentication, you select the option that reflects how you want to use SAML:

- **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, configure Tableau Server to use local authentication rather than Active Directory. 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 local authentication or Active Directory for user management.

SAML compatibility notes and requirements

- **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`.

- **IdP must support forms-based authentication**: You can allow SAML sign-in from Tableau Desktop or Tableau Mobile client applications. To do this, your IdP must support forms-based authentication.

  If it does not, you can disable SAML sign-in for Tableau clients using the `wgserv-er.authentication.desktop_nosaml` command. See `tabadmin` configuration options.

  See also Using SAML SSO with Tableau client applications.

- **Distributed installations**: Clusters configured for SAML must have the same SAML certificate, SAML key, and SAML IdP metadata files on each Tableau Server that runs an Application Server process.

  For more information, see Configure a Server Cluster for SAML.

- **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:

  ```
  http(s)://<tableauserver>/wg/saml/SSO/index.html
  ```

- **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:

  ```
  http(s)://<tableauserver>/wg/saml/SingleLogout/index.html
  ```

- **Post-log-out 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 `tabadmin set wgserv-er.saml.logout.redirect_url` command.
• To specify an absolute URL, use a fully-qualified URL starting with http:// or https://, as in this example:

  `tabadmin set wgserver.saml.logout.redirect_url http://example.com`

• To specify a URL relative to the Tableau Server host, use a page starting with a / (slash):

  `tabadmin set wgserver.saml.logout.redirect_url /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**.

  For configuration information, see Configure SAML with AD FS on Tableau Server.

**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.

**Preserving the Tableau client RelayState value**

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, 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.

XML data requirements

You configure SAML using XML metadata documents that are generated by Tableau Server and by the IdP. During the authentication process, the IdP and Tableau Server exchange authentication information using these XML documents. If the XML does not meet the following requirements, 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.

The SAML metadata XML document that is exported by Tableau Server should contain the following elements, with the Binding attribute set to HTTP-POST.

- Verify the following element which specifies the URL that the IdP redirects to after successful authentication:

  <md:AssertionConsumerService Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST"
  Location="http(s)://TABLEAU-SERVER/wg/saml/SSO/index.html index="0" isDefault="true"/>

- Verify the following element which specifies the URL that the IdP will use for the logout endpoint:
• Verify the following element which specifies the URL for sign in:

<md:SingleSignOnService Binding="urn:oasis:names:tc:SAML:2.0:bindings:HTTP-POST"
Location="http(s)://<tableauserver>/wg/saml/SSO/index.html/>

• **Attribute named `username`:** You must configure the IdP to return an assertion that includes the `username` value 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.

```xml
<saml:Assertion assertion-element-attributes>
  <saml:Issuer>issuer-information</saml:Issuer>
  <Signature xmlns="http://www.w3.org/2000/09/xmldsig#">
    ...
  </Signature>
  <saml:Subject>
    ...
  </saml:Subject>
  <saml:Conditions condition-attributes >
    ...
  </saml:Conditions>
  <saml:AuthnStatement authn-statement-attributes >
    ...
  </saml:AuthnStatement>
```
To change the SAML attribute that passes the username value, use the tabadmin set command. Set the wgserver.saml.idpattribute.username value to the new attribute name (case sensitive). You must change this attribute if you use a global ID.

- 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 Active Directory authentication with Tableau Server and have multiple Active Directory domains (that is, users belong to multiple domains, or your Tableau Server installation includes multiple domains), the IdP must send both the user name 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.

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). If you have multiple sites on Tableau Server and want to use multiple IdP applications, see Configure Site-Specific SAML.


Before you configure Tableau Server for SAML, make sure your environment meets the SAML Requirements.

To configure Tableau Server on Windows for server-wide SAML:

1. Place the SAML certificate files in a folder named SAML, parallel to the Tableau Server 10.5 folder. For example:

   C:\Program Files\Tableau\Tableau Server\SAML

   This is the recommended location because the user account that runs Tableau Server has the necessary permissions to access this folder.

2. In the Tableau Server Configuration Utility, open the SAML tab.

   Post-installation, you can find this utility in the Tableau Server 10.5 > Configure Tableau Server folder.

3. Select SAML authentication for the server and provide the location for each of the following:
**Tableau Server return URL**—The URL that Tableau Server users will be accessing, such as http://tableau_server. Using http://localhost is not recommended. Using 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 like, but it does not have to be your Tableau Server URL.

**SAML certificate file**—A PEM-encoded x509 certificate with the file extension .crt. This file is used by Tableau Server, not the IdP.

**SAML certificate key file**—An RSA or DSA private key file that is not password protected, and that has the file extension .key. This file is used by Tableau Server, not the IdP.

4. Click **Export Metadata File**, and save Tableau Server settings to an XML file with a name of your choice. Later, you will provide this information to your IdP.

   **Note:** If you are configuring SAML for AD FS, instead of going to the next step, continue with the steps described in Step 3: Configure AD FS to accept sign-in requests from Tableau Server, in a separate article.

5. 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 file you exported from Tableau Server earlier in these steps.
   
   - Confirm that your IdP uses **username** as the attribute element to verify.

6. Still within your IdP, export your IdP’s metadata XML file.
It's a good idea to verify that the metadata XML you get from the IdP includes a `SingleSignOnService` element in which the binding is set to HTTP-POST, as in the following example:

```
```

7. Copy your IdP's metadata XML file to same folder on the Tableau Server computer that contains the SAML certificates:

   C:\Program Files\Tableau\Tableau Server\SAML

8. On the SAML tab in the Tableau Server Configuration dialog box, in the SAML IdP metadata file text box, enter the location of the file, and click OK.
Configure a Server Cluster for SAML

When you configure a Tableau Server cluster to use SAML, you place the same SAML certificate, SAML key, and SAML IdP metadata files on every computer that’s running a Tableau application server process (also known as vizportal.exe). To configure a Tableau Server cluster to use SAML:

1. Configure the primary Tableau Server as described in the procedure above.

2. Place the same SAML certificate, SAML key, and SAML IdP metadata files that you used for the primary on each Tableau Worker that is running an application server process. Use the same folder location on the workers that you used on the primary. You do not need to do any additional configuration on the workers.

For example, consider a cluster that includes a primary Tableau Server and two workers. Application server processes run on the primary and on Worker 2 and Worker 3. In this situation, you configure the primary Tableau Server for SAML, and then copy the same SAML certificate, SAML key, and SAML IdP metadata files to the Worker 2 and Worker 3 computers. On the worker computers, put the SAML files in the C:\Program Files\Tableau\Tableau Server\SAML folder, just as they are on the primary computer.

Test Your Configuration

1. In your web browser, open a new page or tab, and enter the Tableau Server URL.

The browser redirects you to the IdP’s sign-in form.
2. Enter your single sign-on user name and password.

![Sign on form]

The IdP verifies your credentials and redirects you back to your Tableau Server start page.

Configure Site-Specific SAML

Use site-specific SAML 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 local authentication and others to use SAML.

Looking for Tableau Server on Linux? See Configure Site-Specific SAML for Tableau Server on Linux.

If you want all server users to use SAML and sign in through the same IdP application, see Configure Server-Wide SAML.

The default authentication method is presented to users that do not belong to a site, that belong to a site but whose server accounts are not configured for single sign-on, or that belong to multiple sites.

To configure Tableau Server to use site-specific SAML, you complete the following high-level steps.
1. Use the Tableau Server Configuration utility to perform one of the following tasks:
   - Configure site-specific SAML with local authentication
   or
   - Configure site-specific SAML with server-wide SAML using local authentication

2. Enable SAML for a site

3. Configure SAML for a site

**Important:** Before you configure Tableau Server for SAML, make sure your environment meets the SAML Requirements. For example, you cannot use Active Directory with site-specific SAML, you must get certificate files, and you must ensure that your IdP provider meets the necessary requirements.

**Configure site-specific SAML with local authentication**

1. If you are configuring SAML during Tableau Server setup, go to the SAML tab in the configuration utility.

   If you are configuring SAML after Tableau Server has been installed, open the Tableau Server Configuration Utility (Start > All Programs > Tableau Server 10.5 > Configure Tableau Server) and then click the SAML tab.

2. On the SAML tab, select the **Site-specific SAML authentication only** option.

3. Place the certificate files that you want to use in a folder named SAML at the same level as the Tableau Server 10.5 folder. For example:

   C:\Program Files\Tableau\Tableau Server\SAML

   You should use this location because the user account that runs Tableau Server has the necessary permissions for accessing this folder.

4. Provide the location for each of the following:
- Tableau Server return URL—The URL that Tableau Server users enter in their browser to access the server, such as http://tableau_server. Using http://localhost will not work for an external-facing server. Using a URL with a trailing slash (for example, http://tableau_server/) is not supported.

- SAML entity ID—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, a site configured for site-specific SAML might display the following entity ID:

  http://tableau_server-samlservice/public/sp/metadata?alias=48957410-9396-430a-967c-75bdb6e002a0

  To find the entity ID, go to the site’s Settings page, and select the Authentication tab. When SAML is enabled, the entity ID is shown in the first step for exporting metadata.

- SAML certificate file—A PEM-encoded x509 certificate (a file with the extension .crt). This file is used by Tableau Server, not the IdP.

- SAML certificate key file—An RSA or DSA private key file that is not password protected, and that has the file extension .key. This file is used by Tableau Server, not the IdP.

5. Click OK.

Continue to Enable SAML for a site.

Configure site-specific SAML with server-wide SAML using local authentication

1. If you are configuring SAML during Tableau Server setup, go to the SAML tab in the configuration utility.
If you are configuring SAML after Tableau Server has been installed, open the Tableau Server Configuration Utility (Start > All Programs > Tableau Server 10.5 > Configure Tableau Server) and then click the SAML tab.

2. On the SAML tab, select the SAML authentication for the server and for sites option.

3. Place the certificate files that you want to use in a folder named SAML that's at the same level as the Tableau Server 10.5 folder. For example:

   C:\Program Files\Tableau\Tableau Server\SAML

   You should use this location because the user account that runs Tableau Server has the necessary permissions for accessing this folder.

4. Provide the location for each of the following:

   • **Tableau Server return URL**—The URL that Tableau Server users enter in their browser to access the server, such as http://tableau_server. Using http://localhost will not work for an external-facing server. Using a URL with a trailing slash (for example, http://tableau_server/) is not supported.

   • **SAML entity ID**—The entity ID that uniquely identifies your Tableau Server installation to the IdP. This is typically the same as the Tableau Server return URL. Additionally, 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, a site configured for site-specific SAML might display the following entity ID:

   http://tableau_server-/saml/service/public/sp/metadata?alias=48957410-9396-430a-967c-75b6e002a0
To find the entity ID, go to the site’s \textbf{Settings} page, and select the \textbf{Authentication} tab. When SAML is enabled, the entity ID is shown in the first step for exporting metadata.

- \textbf{SAML certificate file}—A PEM-encoded x509 certificate (a file with the extension \texttt{.crt}). This file is used by Tableau Server, not the IdP.

- \textbf{SAML certificate key file}—An RSA or DSA private key file that is not password protected, and that has the file extension \texttt{.key}. This file is used by Tableau Server, not the IdP.

5. Leave the \textbf{SAML IdP metadata file} box empty for now and click \textbf{Export Metadata File}.

   A dialog box opens that allows you to save Tableau Server’s SAML settings as an XML file. At this point, metadata from your IdP is not included.

6. Save the XML file. You can give the file any name you want.

7. 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 file that you just exported from Tableau Server.

   - Confirm that your IdP uses \texttt{username} as the attribute element to verify.

8. Still within your IdP, export your IdP’s metadata XML file.

   It’s a good idea to verify that the metadata you get from the IdP includes a \textbf{SingleSignOnService} element in which the binding is set to \texttt{HTTP-POST}, as in the following example:
9. Copy your IdP’s metadata XML file to the following folder on the computer where Tableau Server is installed:

C:\Program Files\Tableau\Tableau Server\SAML

You should use this location because the user account that runs Tableau Server has the necessary permissions for accessing this folder.

10. On the SAML tab in the Tableau Server Configuration dialog box, enter the location of the file in the SAML IdP metadata file box:

11. Click OK. Tableau Server is now configured for SAML authentication.
Continue to the next step, Enable SAML for a site.

Enable SAML for a site

1. Sign in to Tableau Server as a site administrator, and select the site for which you want to enable SAML.

2. Click Settings.

3. Click the Authentication tab.

4. On the Authentication tab, select Use site-specific SAML.

Continue to Configure SAML for a site.

Configure SAML for a site

This section continues from Enable SAML section, and it takes you through each of the configuration steps shown on the Authentication page in Tableau Server.

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:

- 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 generated from the value that you specified when you enabled 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/metadata?alias=48957410-9396-430a-967c-75bdb6e002a0`

- 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.

See the IdP’s SAML configuration steps to confirm the correct option.

**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 name, and then enter the attribute name.

**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.

**Step 7: Troubleshooting**

Start with the troubleshooting steps suggested on the Authentication page. If those steps do not resolve the issues, see Troubleshoot SAML.

**Default authentication type for embedded views**

Part of enabling SAML on your site is to specify how users sign in to access views embedded in web pages.

- **Allow users to choose their authentication type**

  When you select this option, 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** on the Open Web Application Security Project website.
Configure SAML with AD FS on Tableau Server

Active Directory Federation Services (AD FS) allows cloud-based services to use Active Directory (AD) single sign-on (SSO), which is a common method for authentication behind company firewalls. AD FS integrates with SAML, which is an authentication standard available in Tableau Server.

**Note:** These steps reflect a third-party application and are subject to change without our knowledge. If the steps described here do not match the screens you see in your IdP account, you can use the generic SAML configuration steps, along with the IdP’s documentation.

**Prerequisites**

Before you can configure Tableau Server and SAML with AD FS, you must have:

- A server running Microsoft Windows Server 2008 R2 or later, with AD FS 2.0 or later and IIS installed.

- Certificates encrypted using the appropriate encryption method. Be sure to use a SHA-2 (256 or 512 bit) certificate. The last step of this configuration process shows you how to configure Tableau Server for this.

All major browsers will display warnings when connecting to SHA-1 certificates. By the end of 2017, it's likely that most browsers will no longer connect to servers that are presenting SHA-1 certificates.

**Step 1: Configure SSL**

**Note:** If you don’t already have an SSL certificate and key for your Tableau Server, you’ll need to create one. For more information, see Example: SSL Certificate - Generate a Key and CSR. You’ll need to have your certificate signed by a certificate authority (CA). The common name (CN) is used by AD FS and users, so be sure that it is the
fully qualified domain name of your Tableau Server. If there are multiple ways to access the server (e.g., hostname, internal FQDN, external FQDN, etc.), the certificate CN must match the URL that users and AD FS use to talk to Tableau Server.

Step 2: Configure SAML on Tableau Server

Use the SAML tab in the Tableau Server Configuration Utility to enable SAML for Tableau Server and provide the basic required information.

For these steps, see the first several steps in Configure Server-Wide SAML. After you export Tableau Server metadata to an XML file, return here and continue to the next section.

Step 3: Configure AD FS to accept sign-in requests from Tableau Server

Configuring AD FS to accept Tableau Server sign-in requests is a multi-step process, starting with importing the Tableau Server XML metadata file into 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 saml_sp_metadata.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.

<table>
<thead>
<tr>
<th>LDAP Attribute</th>
<th>Outgoing Claim Type</th>
</tr>
</thead>
</table>

- 673 -
Now you will export the AD FS metadata that you’ll import to Tableau Server later, make sure it’s 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.

5. (Optional) This step is required only if AD FS is configured as an IDP for site-specific SAML.

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 (to change the display name to <MySiteName>):

```
Set-ADFSRelyingPartyTrust -TargetName <MySiteName> -EncryptClaims 0
```
Step 4: Provide AD FS metadata to Tableau Server and add support for SHA-2 certificates

1. Return to the Tableau Server Configuration Utility, go to the SAML tab, and for SAML IdP metadata file, enter the location of the XML file you exported from AD FS.

![Tableau Server Configuration](image)

2. Stop Tableau Server, open the Command Prompt as an administrator, and run the following command, so that AD FS can properly redirect to Tableau Server after it accepts users’ SAML credentials:
tabadmin set wgserver.saml.sha256 true

Then start the server again.

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 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 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.

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 Archive Log Files.
**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/`

![Tableau Server Configuration](image)

**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.

**Quick Start: Single Sign-On with Kerberos**

Tableau Server now supports Kerberos-based single sign-on (SSO). Users with Active Directory (AD) accounts in a Kerberos-enabled environment can now use SSO to connect to Tableau Server from Tableau Desktop and web browsers. In addition, Tableau Server can use Kerberos for authentication to Kerberos-enabled Microsoft SQL, MSAS, PostgreSQL, Hive/Impala, and Teradata data sources. When Tableau Server is configured for Kerberos, you can make SSO connections to Cloudera Impala databases using server managed credentials for Impala LDAP authentication.
1 Configure Tableau Server

After you install Tableau Server, run the Tableau Server Configuration utility. On the Kerberos tab select **Enable Kerberos for single sign-on**.

![Tableau Server Configuration](image)

2 Generate the Configuration Script

Click **Export Kerberos Configuration Script** to generate a batch file that will configure Kerberos in AD for Tableau Server.

![Export Kerberos Configuration Script](image)

Save the file and then send it to your AD domain administrator to run.
3 Run the Configuration Script

The domain administrator needs to run the script from a command prompt on any computer in the domain by typing the name of the script.

When your domain administrator runs the configuration script, the script registers Service Principal Names (SPNs) for Tableau Server using the Run As User account, and generates a .keytab file for your environment. (The .keytab file is created in a \keytabs folder in the folder where the script was run.

Have the domain administrator send you a copy of the .keytab file.

4 Copy the .keytab File

On the Kerberos tab of the Tableau Server Configuration utility, enter the path to the .keytab file in the text box in Step 3.

The utility will copy the file to each gateway node in the Tableau Server installation.

Click Test Configuration to verify that the configuration is correctly set up. If the SPNs are correctly set up, the test should display an OK. The number of services configured for delegation will be 0 (zero) unless you have completed the steps below in Configure Kerberos Delegation in AD.

Configure Kerberos Delegation in AD

To use Kerberos Authentication with SQL Server or MSAS data source, or to make SSO connections to Cloudera Impala, you need to configure Kerberos delegation in AD.

You don't need to complete these steps if you will only be using Kerberos SSO to connect to Tableau Server.

To configure Kerberos delegation in AD:
• Enable the Run As User to act as the operating system. For more information, see Enable Run As User to Act as the Operating System.

• Enable Kerberos delegation in AD. This step is specific to the supported connection type(s) that you will be using with Tableau:

  • **SQL Server**—See Enabling Kerberos Delegation for SQL Server in the Tableau Community.
  
  • **MSAS**—See Enabling Kerberos Delegation for MSAS in the Tableau Community.
  
  • **Oracle**—See Enable Kerberos Delegation for Oracle in the Tableau Community.
  
  • **Hive/Impala**—See Enable Kerberos Delegation for Hive/Impala in the Tableau Community.
  
  • **PostgreSQL**—See Enabling Kerberos Delegation for PostgreSQL in the Tableau Community.
  
  • **Teradata**—See Enabling Kerberos Delegation for Teradata in the Tableau Community.

Kerberos Authentication in Tableau Server

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.
User logs into their Active Directory domain.

The Kerberos KDC authenticates the user and sends a Ticket Granting Ticket (TGT) to the user's computer.

The user connects to Tableau Server in Tableau Desktop or in a web browser.

Tableau Server authenticates the user.

Kerberos Requirements

To use Kerberos authentication with Tableau Server, you need the following:

General requirements

- **Windows Server**: Tableau Server must be installed on a server version of Windows. Non-server versions (including Windows 7 and Windows 8) do not support the `ktpass` command required for generating a keytab file.
• **Run As User account:**
  - The Run As User account (the Tableau Server service account) must be an AD domain account. Local accounts, including NTAUTHORITY\NetworkService will not work.
  - The Run As User account must be in the same domain as the database services that will be delegated.
  - Constrained delegation: The Run As User account must be granted access to the target database Service Principal Names (SPNs).
  - Data Source authentication: If you plan to use Kerberos to authenticate to Microsoft SQL Server, MSAS, PostgreSQL or Teradata databases, enable the Run AS User account to act as part of the operating system. For more information, see Enable Run As User to Act as the Operating System.

• **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 Add a Load Balancer and Configure a reverse proxy server for more information.

• **Tableau Mobile app and iOS browser support:** An iOS user can use Kerberos authentication with the Tableau Mobile app and 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.

• **Desktop browser support:** For more information about browser support for Kerberos SSO, see Browser Support for Kerberos SSO.

Active Directory requirements

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](https://support.apple.com/en-us/HT201454).

### Kerberos delegation

For Kerberos delegation scenarios the following are required:

- Kerberos must be configured on Tableau Server. See [Configure Kerberos](#).

- 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 is preferred because it has a dialog for configuring constrained delegation, while 2012 non-R2 requires manual configuration.

### 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 you meet 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.

1. Open a command prompt as an administrator and change directories to the location of Tableau Server’s bin directory. The default location is `C:\Program Files\Tableau\Tableau Server\10.5\bin`.

2. Type the following command to stop Tableau Server:

   `tabadmin stop`

3. Open the Tableau Server Configuration Utility (**Start > All Programs > Tableau Server 10.5 > Configure Tableau Server**), and then click the **Kerberos** tab.

4. Select **Enable Kerberos for single sign-on**.

5. Click **Export Kerberos Configuration Script**. The generated script configures your Active Directory domain to use Kerberos with Tableau Server. For more information, see Kerberos Configuration Script.

   ![Kerberos Configuration](image)

   **Note:** Verify the host names in the setspn lines of the script. If you are using an external load balancer or a reverse proxy, the host names should match the name you used when you configured Tableau Server for the load balancer or proxy. If you have not configured Tableau Server for your proxy or external load balancer, you may need to update the script from the command prompt to match the actual host name.
balancer, do that and then re-export the Kerberos configuration script to ensure it has the correct host names. See Add a Load Balancer and Configuring Proxies for Tableau Server.

6. Have your Active Directory domain administrator run the configuration script to create Service Principal Names (SPNs) and the .keytab file. The domain administrator should do the following:

   - Review the script to verify it contains correct values.
   - Run the script at a command prompt on any computer in the domain by typing the script name (not by double-clicking the script in Windows Explorer).

     The script creates a file, kerberos.keytab, in a \keytabs folder in the location that the script was run.

7. Save a copy of the .keytab file created by the script to the Tableau Server computer. In Step 3, enter the path to the .keytab file, or click the browse button to navigate to the file. The keytab file will be copied to all the gateway nodes in your Tableau Server installation when you click OK in the Configuration utility.

   **Note:** Do not rename the .keytab file. The script creates a file named kerberos.keytab and you need to save it with this name.

8. (optional) Click **Test Configuration** to confirm that your environment is configured correctly to use Kerberos with Tableau Server.
If you have not configured any data sources for Kerberos delegation, 0 is shown for the **Number of services configured for delegation**.

9. Click **OK** to save your Kerberos configuration.

10. Start Tableau Server.

**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:

![Web browser displaying Tableau Server URL](image)

You should be automatically authenticated to Tableau Server.

**Kerberos Configuration Script**

When you click **Export Kerberos Configuration Script** in the Tableau Server Configuration utility, the `KerberosConfig.bat` script is generated. This script registers the Service Principal Names (SPNs) for Tableau Server in Active Directory (AD) and generates a Kerberos `.keytab` file.

**SPNs** - The script uses the `setspn` utility to register the SPNs for Tableau Server, using the Run As User account. These SPNs are used for generating the `.keytab` file, and for authenticating web browser connections to Tableau Server.

`.keytab` - The script uses the `ktpass` utility, to generate a `kerberos.keytab` file, located in the `keytabs` folder in the folder where the script was run. The `.keytab` file contains the shared secret key for Tableau Server.
Note: The setspn and ktpass utilities may generate warning or errors. You can ignore these errors and warnings if the utilities run to completion.

Enable Single Sign-On for SAP BW

Starting with Tableau 10.2, you can configure single sign-on (SSO) support from Tableau to SAP NetWeaver Business Warehouse (SAP BW). To use this feature, you must install and configure software on the computers that run Tableau Server and Tableau Desktop. Detailed instructions can be found on Tableau Community:

- To configure impersonation from Tableau Server to SAP BW using server side trust, see Configuring Delegation from Tableau Server to SAP BW on Tableau Community. Note that Tableau Server doesn't use Kerberos to connect to SAP BW, so you don't configure Kerberos on Tableau Server.

- To configure Kerberos authentication from Tableau Desktop to SAP BW, see Configuring Single Sign-On for Tableau Desktop to SAP BW on Tableau Community.

Browser Support for Kerberos SSO

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 permission to access Tableau Server (they can log in using a user name and password)
- The user must be authenticated to Active Directory through Kerberos on the client computer. Specifically, this means that they have a Kerberos Ticket Granting Ticket (TGT).

Note: If Kerberos SSO fails you can still sign in to Tableau Server with your user name and password.
Desktop clients

The following combinations of operating system and Tableau Desktop or browser support Kerberos authentication to Tableau Server. In some cases, additional configuration is required.

Windows

- Tableau Desktop 8.3 or newer
- 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

Mobile clients

The following combinations of operating system and Tableau App or browser support Kerberos authentication to Tableau Server:

MaciOS

- 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
Note 1: Internet Explorer and Google Chrome on Windows desktop

Kerberos SSO is supported on both Internet Explorer and Google Chrome but requires configuration in Windows Internet Options:

- Enable Integrated Windows Authentication.
- Verify that Tableau Server URL is in the local intranet zone. The Tableau Server URL must be in the local internet zone. In many cases Internet Explorer’s automatic detection of intranet zones will automatically configure this. If the URL has not been detected and configured, then you will need to 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.
4. The internal Tableau Server URL should be listed in the Websites field. In some organizations, IT administrators will use a wildcard (*) to specify internal URLs. For example, the wildcard, https://*.example.lan, includes all servers in the internal example.lan namespace in the local intranet zone.
If the Tableau Server URL or a wildcard URL for the Tableau Server namespace is listed in the **Websites** field, then click **Close**. You do not need to update the local intranet zone.

If the Tableau Server URL or a wildcard URL is not specified in the **Websites** field, Enter the Tableau Server name (URL) in the **Add the website to the zone** field, click **Add**, and then click **OK**.

**Note 2: Firefox on Windows and Mac OS X desktop**

Kerberos SSO is supported in Firefox on both Windows and Mac, but requires that you configure two URL settings to enable Kerberos. Both of those settings are described in the following procedure:

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` or `example.com`.

**Note 3: Chrome on Mac OS X desktop**

According to Chrome documentation, Kerberos SSO works a Mac when you launch Chrome from a terminal window with the following command:

```
open -a "Google Chrome.app" --args --auth-server-whitelist="tableau-server.example.com"
```

where `tableau-server.example.com` is the URL for Tableau Server in your environment.

However, we have found inconsistent results in our testing and therefore recommend using Safari or Firefox if you want to use Kerberos SSO on a Mac. See the Integrated Authentication section at [HTTP authentication](#) on The Chromium Projects site for more information.
**Note**: Chrome on Mac OS X continues to be supported with Tableau Server, but users may be prompted to log in with their user name and password (single sign-on may not work).

**Note 4: Mobile Safari and Tableau App 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 App or a supported browser to connect to and log 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**

**Kerberos Authentication Failed (unable to connect automatically to Tableau Server)**

If you are using Kerberos for SSO and a user is prompted to sign in to Tableau Server when they connect with either a web browser or with Tableau Desktop, try these steps from the client computer:
Troubleshooting on the client computer

- **Account permissions**—Try to sign in to Tableau Server using the user’s name and password. If they can’t sign in to Tableau Server using their user name and password, they do not have permission to access Tableau Server and Kerberos authentication will fail.

- **Other accounts**—Try to connect with SSO to Tableau Server using other user accounts. If all users are affected, the problem may be in the Kerberos configuration.

- **Computer location**—Kerberos will not work when connecting from localhost. Clients must be connecting from a computer other than the Tableau Server computer.
• **URL address**—You cannot use Kerberos SSO when connecting using an IP address. 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).

• **TGT (Ticket Granting Ticket)**—Confirm that the client computer has a TGT from the Active Directory domain. Kerberos requires a TGT to sign in. Constrained delegation, with the proxy granting a ticket, is not supported.

For more information, see Single-Sign On (SSO) in Kerberos Requirements.

To confirm the client computer has a TGT, type:

- `klist tgt` at a command prompt on a Windows computer
  - or
- `klist` at a terminal prompt on a Mac computer

The output should show a TGT for the user/domain trying to authenticate to Tableau Server.

The client computer may 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 into the computer with a local (non-domain) account
- The computer is a Mac that is not using Active Directory as a network account server

• **Browser**—Check which browser the user is using to access the server

  - Internet Explorer (IE) and Chrome work "out of the box" on Windows
  - Safari works "out of the box" on Mac
Firefox requires additional configuration

For more information about browser support for Kerberos Single Sign-On (SSO), see Browser Support for Kerberos SSO.

Troubleshooting 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.
  
  - In a ziplog archive, these logs are in the `httpd` folder.
  
  - On Tableau Server, these logs are in the `\data\tabsvc\logs\httpd\` folder.

- **Key table entry**—If the error.log entry says "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_cred() failed: Unspecified GSS failure. Minor code may provide more information, No key table entry found matching HTTP/servername.domain.com@)

  This error is a result of a mismatch between 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 Tableau Server computer name needs to match in:

- .keytab file
- Service Principal Name (SPN) for the server

**Test Configuration and tabconfig.log**

Use the Test Configuration button in the Tableau Server Configuration utility:

<table>
<thead>
<tr>
<th>Test Configuration</th>
<th>Tests For:</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>* SPNs are correctly configured: OK</td>
</tr>
<tr>
<td></td>
<td>* Number of services configured for delegation: 0</td>
</tr>
</tbody>
</table>
If your SPNs are correctly set up for Kerberos, **SPNs are correctly configured** shows OK.

If any services are configured for delegation, the number of configured services will appear. A value of 0 (zero) does not indicate a problem unless you are using delegation and Kerberos authentication to SQL Server or MSAS.

Look in `tabconfig.log` for any problems or errors. For example:

```
2014-10-17 10:58:16.545 -0700 ERROR root: No SPN entries found
```

If the test does not show successful results, run the configuration script again.

**Data source SSO**

**Delegated data source access failures**

Check the `vizqlserver` log files for "workgroup-auth-mode."

- In a ziplog archive, these logs are in the `\vizqlserver\Logs` folder

- On the Tableau Server, these logs are in the `\data\tabsvc\vizqlserver\Logs` folder

Look for "workgroup-auth-mode" in the log files. It should say "kerberos-impersonate" not "as-is".

**Kerberos delegation multi-domain configuration**

Tableau Server has the ability to 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\j smith`, whose Kerberos Principal is `j smith@EXAMPLE.LAN`. In this case, Tableau Server will specify a delegated user, `j smith@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)s/@.*//
    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)s/@.*//
    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 wish to preserve an existing single domain delegation alongside cross-domain delegation, then you must configure both using resource-based constrained delegation.

The article, How Windows Server 2012 Eases the Pain of Kerberos Constrained Delegation, Part 2, provides a detailed description of how Microsoft KDC and Active Directory handle delegated scenarios. To configure these scenarios, refer to the Microsoft Kerberos documentation.

**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 requests a resource on Tableau Server.

Step 2: Tableau Server redirects the request to the IdP gateway for authentication.

Step 3: The user is prompted and successfully authenticates with the IdP. The IdP redirects the session back to Tableau Server. Included with the URL is an authorization code for the user.

Step 4: Tableau Server presents the user's authorization code back to the IdP. As an IdP client, the Tableau Server includes its own client credentials to prevent spoofing.

Step 5: The IdP returns an access token and an ID token to Tableau Server.

- The access token contains an encrypted key to verify that the IdP isn't being spoofed. The process the key is used for is referred to as JSON Web Token (JWT) validation. The key for JWT is specified in the jwks_uri in the IdP discovery document. If your IdP does not support JWT validation, then you can disable it. See "viz-portal.openid.ignore_jwk" in the tabadmin configuration options topic.

- 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" : 


"7gYhRR3HiRRCaRcgvY50ubrtjGQBMr4W4rXbpPFp-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 6: Tableau identifies user from IdP claim. Tableau Server searches the user account records stored in the repository. By default, Tableau Server will use the subject identifier, or sub, claim to identify a user account. If no user record is storing the sub claim value, then Tableau Server will search for username matches using the email claim. When a username match is found, Tableau Server will write the corresponding sub claim to the user record in the repository. You can configure Tableau Server to use different claims for this process. See Requirements for Using OpenID Connect.

Step 7: Tableau Server authorizes the user.

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, use the following sequence of `tabadmin` commands:

```
tabadmin stop
    tabadmin set vizportal.openid.ignore_domain true
```
tabadmin configure
tabadmin start

When you change the `vizportal.openid.ignore_domain` setting 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 `sub` claim is often included in IdP claims. Typically, the `sub` claim is a unique string that identifies a given user account. The benefit of using a `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 `sub` claim in the IdP ID token.

The OpenID `sub` claim value must be mapped to the corresponding user in Tableau Server. Since the `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, `email`, to identify the Tableau user. Tableau will then update that user’s record with the `sub` claim from OpenID. Since the ID token always includes the `sub` claim along with other claims, on subsequent sessions, Tableau will identify that user with the `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 `kwilliams`. 

- 707 -
To change the IdP claim that is used to map identity on Tableau Server, use the following sequence of `tabadmin` commands:

```
tabadmin stop
  tabadmin set vizportal.openid.username_claim arbitraryClaim
  tabadmin configure
  tabadmin start
```

Where `arbitraryClaim` is the name of the IdP claim that you want to use for mapping OpenID accounts to existing Tableau Server usernames.

**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 following sequence of tabadmin commands:

```
  tabadmin stop
  tabadmin set vizportal.openid.id_claim arbitraryClaim
  tabadmin configure
  tabadmin start
```

Where arbitraryClaim is the name of the IdP claim that you want to use as the replacement for the sub claim.

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 "vizportal.openid.ignore_jwk" in the tabadmin
configuration options topic.

- Tableau Server expects a \texttt{kid} value in the \texttt{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 \texttt{jwks_uri} value in the OpenID discovery document. A \texttt{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 \texttt{x5c} parameter or for using X.509 certificates.

For more information about OpenID Connect, see the following:

- OpenID Connect Core 1.0 incorporating errata set 1
- OpenID Connect Discovery 1.0 incorporating errata set 1

Configure the Identity Provider (IdP) 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 (IdP) for OpenID Connect (you are here)
- Configure Tableau Server for OpenID Connect
- Signing In to Tableau Server Using OpenID Connect
- Changing IdPs in Tableau Server for OpenID Connect

Configure the IdP

Before you can use OpenID Connect with Tableau Server, you must have an account with an 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.

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.

   The IdP configuration requires an additional step that you cannot finish until after you’ve configured Tableau Server, as described in Configure Tableau Server for OpenID Connect.
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 (IdP) for OpenID Connect
- Configure Tableau Server for OpenID Connect (you are here)
- Signing In to Tableau Server Using OpenID Connect
- Changing IdPs in Tableau Server for 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 (IdP) for OpenID Connect.

Important notes

Before you configure Tableau Server for OpenID Connect, make sure you read these notes.

- You can use OpenID Connect with Tableau Server only if the server is configured to use local authentication. OpenID Connect is not available if the server is configured to use Active Directory authentication. For more information, see Configure General Server Options.

- If you are configuring Tableau Server for the Salesforce IdP, then you must set the `vizportal.openid.client_authentication` parameter. See `tabadmin` configuration options for more information.

- We recommend that you configure Tableau Server to use SSL for external communications. This helps to maintain secure communications between Tableau Server
and the IdP during the exchange of authentication information. For details, see Configure External SSL.

If your IdP uses a self-signed certificate (or the IdP certificate is not signed by a trusted CA), then you can add the IdP public certificate to the Java truststore. See Configuring Tableau Server for OpenID Connect Using Uncommon or Self-Signed Certificates.

If you are configuring OpenID Connect during the initial configuration of Tableau Server (the first time the configuration utility runs), there is no option to set up SSL. In that case, we recommend that you finish the installation, then return to the configuration to set up SSL and then configure OpenID.

**Note** If you want to use external SSL for Tableau Server, it's generally more convenient to do that before you configure OpenID Connect. If you configure SSL after you've already configured OpenID, you need to return to the IdP and update the configuration that you made previously. For example, you need to change the protocol for the Tableau Server external URL from `http://` to `https://`.

Configure the server

To configure Tableau Server for OpenID Connect, follow these steps.

1. Log in as an administrator to the computer where Tableau Server is running.

2. If the server is running, stop it (Windows Start > **All Applications** > **Tableau Server** > **Stop Tableau Server**).

   **Tip:** You can also stop the server by using the `tabadmin stop` command.

3. Run the Tableau Server Configuration tool (Windows Start > **All Applications** > **Tableau Server** > **Configure Tableau Server**).
4. Click the **OpenID** tab.

5. Select the **Use OpenID Connect for single sign-on** option.

6. Fill in the **Provider client ID** and **Provider client secret** boxes with the values you recorded earlier.

7. In the **Provider configuration URL** box, enter the URL that the IdP uses for OpenID Connect discovery.

   Alternatively, you can configure Tableau Server to reference a static discovery document to configure OpenID. The discovery file is a JSON document that defines the IdP configuration. Copy the file to the Tableau Server and then run the following sequence of `tabadmin` commands:

   ```
   tabadmin stop
   tabadmin set vizportal.openid.static_file <file-path>
   tabadmin configure
   tabadmin start
   ```

   Where `<file-path>` is the full path of the discovery file, for example, `c:\files\openid-configuration`.

8. In the **Tableau Server external URL** box, enter the URL of your server. This is typically the public name of your server, such as `http://example.tableau.com`.

   When you initially configure OpenID, the **Provider configuration URL** box contains a default value that's 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.

   **Note:** Make sure that you update the external URL if the default value is not the URL for how your server can be reached from an external source.
9. Copy the URL in the box labeled **Configure the OpenID provider using the following redirect URL for Tableau Server**. You’ll use this value in the next procedure to finish configuring the IdP.

10. Start the server (Windows Start > **All Applications** > **Tableau Server** > Start **Tableau Server**).

    **Tip:** You can also start the server by using the tabadmin start command.

Add the redirect URL to the IdP configuration

After you configure Tableau Server, you finish the IdP configuration using the server’s redirect URL.

1. Return to the IdP portal where you set up the project or application.

2. Edit the project configuration and find the redirect URL.

3. Enter the redirect URL that you copied in the previous procedure.
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 (IdP) for OpenID Connect
- Configure Tableau Server for OpenID Connect
- Signing In to Tableau Server Using OpenID Connect (you are here)
- Changing IdPs in Tableau Server for OpenID Connect

Signing in using OpenID Connect

Once 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, tabadmin, 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 `tabadmin set wgserver.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, do the following:

1. Stop the server.
2. Run the following sequence of `tabadmin` commands:
   
   ```bash
   tabadmin set wgserver.authentication.restricted true
   tabadmin configure
   ```
3. Start the server.

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. The following topics provide information about configuring and using OpenID Connect with Tableau Server.

- OpenID Connect
- Configure the Identity Provider (IdP) for OpenID Connect
- Configure Tableau Server for OpenID Connect
- Signing In to Tableau Server Using OpenID Connect
- Changing IdPs in Tableau Server for OpenID Connect (you are here)

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 `tabadmin reset_openid_sub` command. You can reset (that is, clear) sub values for an individual user, as in the following example:

`tabadmin reset_openid_sub --username Alice`

You can also clear the sub value for all users using this command:

`tabadmin reset_openid_sub --all`

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 on the list above, 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 tabadmin command:

```
tabadmin set vizportal.openid.custom_scope 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).

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 and you have configured OIDC for the Salesforce IdP, then you must set the \texttt{vizportal.openid.client\_authentication} parameter. See \texttt{tabadmin} configuration 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.
can create a snapshot of log files and use them to troubleshoot problems. For more information, see Archive Log Files.

**Note:** To log OpenID-related events, `vizportal.log.level` must be set to debug. For more information, see Change Logging Levels.

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.

1. **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.

2. **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.

3 **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 `tabadmin` configuration options.
How is a trusted ticket stored?

As of version 10.2, 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:

1. Open a command prompt as an administrator and navigate to your Tableau Server bin directory (for example, C:\Program Files\Tableau\Tableau Server\10.5\bin).
2. Type the following command to stop Tableau Server:
tabadmin stop

3. Next, type the following command:

```
tabadmin set wgserver.trusted_hosts "<trusted IP addresses or host names>
```

In the command above, `<trusted IP addresses>` should be a comma-separated list of the IPv4 addresses or host names of your web server(s).

**Note:** The values you specify completely overwrite any previous setting. Therefore, you must include the full list of hosts in the `set` command. (You cannot amend the list of hosts by running the `set` command repeatedly.)

For example:

```
tabadmin set wgserver.trusted_hosts "192.168.1.101, 192.168.1.102, 192.168.1.103"
```

or

```
tabadmin set wgserver.trusted_hosts "webserv1, webserv2, webserv3"
```

**Notes:**
The comma separated list should be in quotes, with one space after each comma.
The web servers you specify must use static IP addresses, even if you use host names ([learn more]).

4. 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 How Trusted Authentication Works) and Tableau Server, you also need to add them as trusted gateways.
See Configure a reverse proxy server for steps.

5. Type the following command to save the changes to all the server configuration files:

   `tabadmin config`

6. Finally, type the following command to start the server again:

   `tabadmin start`

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://t-bserv`. For example `http://tabserv/trusted`.

**Note:** If SSL is enabled you must use `https` instead of `http`. For example: `https://tabserver/trusted`.

For code examples that you can use to create the POST request in Java, Ruby, and PHP, see the following:

C:\Program Files\Tableau\Tableau Server\10.5\extras\embedding

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 user name (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=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 has changed in Tableau Server 10.5. 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: 9D10byqDQmSI0yQpKdy4Sw==:dg62gCsSE0QRaXNTQp6mlJ5.

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://tabserver/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:

```
<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"
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="9D1ObyqDQmSI0yQpKdy4Sw==: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:

```html
<script type="text/javascript" src="http://myserver/javascripts/api/viz_v1.js"></script>
<object class="tableauViz" width="900" height="700" style="display:none;">
  <param name="path" value="/true-sted/9D1ObyqDQmSI0yQpKdy4Sw==:dg62gCsSE0QRArXNTOp6mlJ5/views/MyCoSales/SalesScoreCard" />
</object>
```

Here’s the same example, but for a multi-site server. Note that /t/<site ID> is used here:

```html
<script type="text/javascript" src="http://myserver/javascripts/api/viz_v1.js"></script>
<object class="tableauViz" width="900" height="700"
```
Iframe Tag Example

<iframe src="http://tabserver/trusted/9D1ObyqDQmSI0yQpKdy4Sw==:dg62gCsSE0QRaXNTOp6mlJ5/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 command window and change directories to the location of Tableau Server’s `bin` directory. The default location is `C:\Program Files\Tableau\Tableau Server\10.5\bin`.

2. Open a command prompt as an administrator and type the following command:

```
tabadmin set wgserver.extended_trusted_ip_checking true
```

3. Then type the following command:

```
tabadmin configure
```

4. Finally, restart the server by typing the following:

```
tabadmin restart
```
Testing Trusted Authentication

The steps below detail a method for testing the retrieval of 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.

Step 1: Add web server IP address to wgserver.trusted_hosts tabadmin option

1. On the Tableau Server, open Command Prompt and Run as administrator.
2. In Command Prompt, navigate to the Tableau Server bin directory. For example:

   cd "C:\Program Files\Tableau\Tableau Server\Tableau Server-\bin"

3. Specify the IP address of the trusted web server that will be proxying authentication requests to Tableau Server by running the following commands:

   tabadmin set wgserver.trusted_hosts <ip-address>
   
   tabadmin config
   
   tabadmin restart

   See tabadmin configuration options for more information about setting wgserver.trusted_hosts.

Step 2: Add a test user

Create a user on the Tableau Server that you can use to test trusted ticket functionality. See Add Users to the Server. Set the role of the user to Interactor.

Step 3: Create a test HTML page

Copy the following code and save as an html file on the web server that you specified in Step 1.

<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>
<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="http://" />
</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>
<td class="style2">
Site: (leave blank for Default site, else NameOfSite if using sites)</td>
</td>
</tr>
<tr>
<td class="style2">
<input type="text" id="target_site" name="target_site" value="" />
</td>
</tr>
<tr>
<td class="style2">
<input type="submit" name="submittable" value="Go" />
</td>
</tr>
</table>
</form>

<H4>Be sure to add your IP as a Trusted IP address to the server</H4>

Step 4: 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 Step 3. The web browser may prompt you to allow scripts to run. This operation requires Javascript.
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 (only if configured for client trusted IP matching).
   - **Site**: The name of the Tableau Server site where test user is hosted.

3. Click **Go**. 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,
     9D101xmDQmSIoYQpKdy4Sw==:dg62gCsSE0QArXNTOp6mlJ5.
   - **-1**: If the value, -1 is returned, then there is an error in the configuration. See Ticket Value of -1 Returned from Tableau Server.

Step 5: 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 Step 4 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://<Tableau Server name>/trusted/<unique ticket from Step 4>/views/<workbook>/<view>

Non-default site server url

https://<Tableau Server name>/trusted/<unique ticket from Step 4>/t/<site name where test user is hosted>/views/<workbook>/<view>

Variables in the URLs are encapsulated with 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 ProgramData\Tableau\Tableau Server\data\tabsvc\logs\vizqlserver\vizql*.log. To increase the logging level from info to debug, use the tabadmin setting vizqlserver.trustedticket.log_level.

To test your trusted authentication deployment, see Testing 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. The exact reason for this message is written to the vizql*.log files in the following folder:

ProgramData\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 IP address or host name for the computer sending the POST request must be 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.
• Value of wgserver.trusted_hosts is properly formatted

The list of trusted hosts you provided using the `wgserver.trusted_hosts` setting must be a comma-separated list with a space after each comma. For example, the list should be similar to the following: 192.168.1.101, 192.168.1.102, 192.168.1.103, or bigbox1.example.lan, bixbox2.example.lan, bigbox3.example.lan.

• 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: 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 with a Viewer or Interactor license level. You can see a list of users and their license levels by signing in to Tableau Server as an administrator and clicking the Licensing link on the left side of the page.

• 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`.

• 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 Enable automatic login.

If you see a 401 error (or a 302 - Redirect error) after you have deployed Tableau Server 10.5, 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.

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 tabadmin set option vizqlserver.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 set to change it to false.
Data Connection Authentication

More information

- 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.

Tableau Server requires constrained delegation, with the Run As User account specifically granted delegation rights to the target database Service Principal Names (SPNs). Delegation is not enabled in Active Directory by default.

**Note:** Kerberos constrained delegation for SSO to Tableau Server is not supported. Constrained delegation for data sources is supported. The client computer must have a TGT from the Active Directory domain. For more information, see Single-Sign On (SSO) in Kerberos Requirements.

To configure Kerberos delegation:

2. On all nodes in Tableau Server, configure the Run As User to act as part of the operating system. For more information, see Enable Run As User to Act as the Operating
System.

3. In Active Directory:
   - Configure SPNs for the data sources you will be using.
   - Enable Kerberos delegation for the data sources’ SPNs
   - (Optional for multi-domain environments) Configure krb5_conf.html to map principal names to local user names for each Kerberos realm. See Kerberos delegation multi-domain configuration.

4. Enable delegation for data connections:
   - **SQL Server**—See Enabling Kerberos Delegation for SQL Server in the Tableau Community.
   - **MSAS**—See Enabling Kerberos Delegation for MSAS in the Tableau Community.
   - **Oracle**—See Enable Kerberos Delegation for Oracle in the Tableau Community.
   - **Hive/Impala**—See Enable Kerberos Delegation for Hive/Impala in the Tableau Community.
   - **PostgreSQL**—See Enabling Kerberos Delegation for PostgreSQL in the Tableau Community.
   - **Teradata**—See Enabling Kerberos Delegation for Teradata in the Tableau Community.

Enable Run As User to Act as the Operating System

To use Kerberos delegation with Tableau Server, you must configure the Run As User account to act as the operating system on each Tableau Server node.
1. On the computer that is running Tableau Server, select **Start > Control Panel > Administrative Tools > Local Security Policy.**

2. In the Local Security Settings window, expand **Local Policies**, click **User Rights Assignments**, and then right-click **Act as part of the operating system** and select **Properties.**

3. In the **Act as part of the operating system** Properties window, click **Add User or Group.**

4. Type the `<domain>\<username>` for the Tableau Server Run As User account (for example: `MYCOMPANY\tableau_server`), and then click **Check Names.**
5. When the account resolves correctly, it is underlined. Click **OK**.

6. Click **OK** to close the Local Security Policy windows.

**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 Server 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.

- 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 two procedures:

- Get required information from Google and enable API access.
- Use the information you obtained to configure your server.
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.
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:
1. On the Tableau Server computer, open the Command Prompt as an administrator and change to the Tableau Server bin directory.

   cd C:\Program Files\Tableau\Tableau Server\<version>\bin

2. Type the following command to stop the server:

   tabadmin stop

3. Type the following commands to configure the server with the client ID and client secret you obtained from Google, as well as your server URI. Press Enter after each command.

   tabadmin set oauth.google.client_id <your_client_ID>

   tabadmin set oauth.google.client_secret <your_client_secret>

   tabadmin set oauth.google.redirect_uri <your_authorized_redirect_URI>

4. Type the following commands to complete the configuration and restart the server:

   tabadmin config

   tabadmin start

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.
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:
Configure Tableau Server for Salesforce.com OAuth

1. On the Tableau Server computer, open the Command Prompt as an administrator and change to the Tableau Server bin directory:

   ```bash
   cd C:\Program Files\Tableau\Tableau Server\<version>\bin
   ```

2. Type the following command to stop the server:

   ```bash
   tabadmin stop
   ```

3. Type the following commands to configure the server with the consumer ID and secret you obtained from Salesforce and the callback URL. Press Enter after each command:

   ```bash
   tabadmin set oauth.salesforce.client_id <your_consumer_ID>
   tabadmin set oauth.salesforce.client_secret <your_consumer_secret>
   tabadmin set oauth.salesforce.redirect_uri <your_callback_URL>
   ```

4. (Optional) To change the default login server, type the following command:

   ```bash
   tabadmin set oauth.salesforce.server_base_url <URL>
   ```

   By default, this is set to https://login.salesforce.com.

5. Type the following commands to complete the configuration and restart the 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 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

1. On the Tableau Server computer, open a command prompt as an administrator and change to the Tableau Server bin directory using the following command:

   ```
   cd C:\Program Files\Tableau\Tableau Server\<version>\bin
   ```

2. Type the following command to stop the server:

   ```
   tabadmin stop
   ```

3. Type the following commands to configure the server with the app token, consumer key, and consumer secret that you copied from the Intuit site.

   ```
   tabadmin set oauth.quickbooks.oauth_callback_uri http://YOUR-SERVER/auth/add_oauth_token
   tabadmin set oauth.quickbooks.consumer_key <your_consumer_key>
   tabadmin set oauth.quickbooks.consumer_secret <your_consumer_secret>
   ```

4. Type the following commands to complete the configuration and restart the server:

   ```
   tabadmin config
   tabadmin start
   ```
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 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.

**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. **Single-site:** Click Settings > General.
   - **Multisite:** In the site menu, click Manage All Sites and then click Settings > General.

3. In the 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

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.

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:
• 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

Configure SAP HANA SSO

You can configure Tableau Server to use SAML delegation to provide Single Sign-on (SSO) for SAP HANA. HANA SSO is not dependent on SAML authentication to Tableau Server. Note: You do not need to use SAML sign on with Tableau Server in order to use HANA SSO. You can sign in to Tableau Server using whatever method you choose.

With SSO for SAP HANA, Tableau Server functions as an Identity Provider (IdP) and this configuration allows you to provide a single sign-on experience for users making SAP HANA connections. As part of the configuration, you need to acquire a SAML certificate and key file for Tableau Server (these should be a public key certificate and private key). You need to also install the signed certificate in HANA. You can generate the certificate and key yourself, or get them from a Certificate Authority. For more information on generating a certificate/private key and configuring SAP HANA, see the Tableau Community.
Note: The SAP HANA driver version 1.00.9 or later must be installed on Tableau Server in order to use SSO for SAP HANA. The driver cannot encrypt the SAML assertion, so you may want to enable encryption for the SAML connections. For more information, see the Tableau Community.

Configure SSO for SAP HANA

To configure Tableau Server to use SSO for SAP HANA:

1. Place certificate files in a folder named SAML, parallel to the Tableau Server 10.5 folder. For example:

   C:\Program Files\Tableau\Tableau Server\SAML

   You should use this location because the user account that runs Tableau Server has the necessary permissions for accessing this folder.

2. After you install Tableau Server, run the Configuration utility (Start > All Programs > Tableau Server 10.5 > Configure Tableau Server), and then click the SAP HANA tab.

3. Select Use SAML to enable single sign-on for SAP HANA and provide the location for each of the following:

   **SAML certificate file**—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.

   **SAML private key file**—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.
4. Select the format of the user name.

5. Select the case for the user name. This determines the case of the name when it is forwarded to the SAP HANA identity provider (IdP).

**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. In other words, if Jane Smith’s Tableau Server user account has a username of MyCo\jsmith and Tableau Server is using Active Directory for user authentication, her username on the SQL Server database must also be MyCo\jsmith and 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). This SQL Server account must also be one of two accounts on the Tableau side of things:
- The Tableau Server Run As User account (see Impersonate with a Run As User Account).

- 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 User Account

Impersonating via a Run As User account is the recommended way to perform impersonation. The Run As User account is an Active Directory (AD) account the Tableau Server service can run under on the machine hosting Tableau Server (see Run As User). 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 account for impersonation gives the administrator the most control.

To set up impersonation with a Run As User account:

1. When you configure Tableau Server as part of Setup, under Server Run As User, enter the Run As User AD account that has IMPERSONATE permission for the user accounts. Under User Authentication, select Use Active Directory:
2. Click **OK** to finish configuration.

3. 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:

![SQL Server Connection](image)

4. In Tableau Desktop, publish the workbook to Tableau Server (**Server > Publish Workbook**).

5. 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:
6. Click **OK**.

7. 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 are permission sets that are assigned to a user, such as System Administrator, Publisher, or Viewer. The site roles define collections of capabilities (delete, save, view, and others) that can be granted to users or groups on Tableau Server.

Site roles define who is an administrator. Administrators can be assigned at the site or server level. Site roles also determine whether non-admin users are allowed to publish to the server from Tableau Desktop. In general, site roles determine the maximum capabilities that can be granted for each non-admin user. For example, if a user’s site role is Interactor, the user cannot publish to the server, no matter what other permissions the user has, because the Interactor role denies permission to publish.

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 resource.

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 resources 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 resource-based. Permissions are assigned to individual resources and are granted to users or groups. Permissions are evaluated for projects, workbooks, views, and data sources.

- Permissions are implicitly denied, and non-admin users must explicitly be allowed to access resources. The process by which Tableau Server determines the “allow” or “deny” permission is explained in detail in the topic, How Permissions are Evaluated.

- Permissions inheritance exists only in locked projects and in workbooks with tabbed views. When content permissions are locked to the project, its workbooks, views, and data sources will always use the default permissions in the project. In the case of workbooks saved with the option Show sheets as tabs, views will use the workbook permissions. For more information, see Content Permissions and Ownership.

- In a project that is not locked, initial permissions are a one-time copy of the container item’s permissions. A workbook, view, or data source will start with the default permissions, but authorized users can subsequently edit permissions on those resources. For more information on default permissions and projects, see Set Project Default Permissions and Lock the Project.

Tableau Server provides a flexible permissions infrastructure that allows you to manage access to all content for countless scenarios. See Content Permissions and Ownership for more detailed information.

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 may require authorization that is outside the scope of Tableau Server’s authority. If users publish an external data source, then Tableau Server will manage access and capabilities of data source. But if users embed an external data source in a workbook, then it’s up to the users who publishes the workbook to determine how other users who open the workbook will authenticate with the data source.

- 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 may 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.

Server Administration: Authorization for Configuring Tableau Server

One or more users must have Windows local admin permissions to configure Tableau Server and to run tabadmin configuration options commands.

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><strong>Database Connection Options</strong></th>
<th><strong>Data Security Questions</strong></th>
</tr>
</thead>
<tbody>
<tr>
<td>Database login account uses...</td>
<td>Is database security possible per Tableau Server user?</td>
</tr>
<tr>
<td><strong>Window NT Integrated Security (Windows Authentication)</strong></td>
<td></td>
</tr>
<tr>
<td>Server Run As account</td>
<td>No</td>
</tr>
<tr>
<td>Impersonate via server Run As account</td>
<td>Yes</td>
</tr>
<tr>
<td>Viewer Credentials</td>
<td>Yes</td>
</tr>
<tr>
<td><strong>Username and Password</strong></td>
<td>Prompt user: Viewers are prompted for their database credentials when they click a view. Credentials can be saved.</td>
</tr>
<tr>
<td>Embedded credentials: The work-book or data source publisher can embed their database credentials.</td>
<td>No</td>
</tr>
<tr>
<td>Impersonate via embedded password: Database</td>
<td>Yes</td>
</tr>
<tr>
<td><strong>Database Connection Options</strong></td>
<td><strong>Data Security Questions</strong></td>
</tr>
<tr>
<td>---------------------------------</td>
<td>-----------------------------</td>
</tr>
<tr>
<td>Database login account uses...</td>
<td>Authentication mode</td>
</tr>
<tr>
<td></td>
<td>Is database security possi-</td>
</tr>
<tr>
<td></td>
<td>ble per Tableau Server user?</td>
</tr>
<tr>
<td></td>
<td>Are user filters the only way</td>
</tr>
<tr>
<td></td>
<td>to restrict which data each</td>
</tr>
<tr>
<td></td>
<td>user sees?</td>
</tr>
<tr>
<td></td>
<td>Are web caches shared among</td>
</tr>
<tr>
<td></td>
<td>users?</td>
</tr>
<tr>
<td>credentials with impersonate</td>
<td></td>
</tr>
<tr>
<td>permission are embedded.</td>
<td></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.

Some of the options described above require configuration steps that must happen during Tableau Server Setup or before you publish a workbook or data source. See the following topics for more information:

- Server Settings (General)
- Enable Kerberos Delegation
- OAuth Connections
- Run As User
- SQL Server Impersonation
- User Filters and Data Source Filters in the Tableau Help.

Related Topics

Regenerate a Password for the Tableau Server PostgreSQL Database (Repository)

Manage Server Secrets

Regenerate a Password for the Tableau Server PostgreSQL Database (Repository)

When you install Tableau Server or upgrade from a previous version, the installation process generates a password for Tableau Server to use internally when it accesses the Repository PostgreSQL database. To help with security, the password generated during the installation process is unique to an installation. Because the password is used only by Tableau Server for access to the Repository, the password is not accessible to server administrators or other users.

Tableau Server can also generate an SSL certificate that can be used to protect internal communications to the Repository and other server components. Using SSL for internal communications between processes is optional. For more information, see Configure Internal SSL.

Note: If you need access to the Repository (for example, to monitor activity), you can use the administrative views that are built in to the server environment or create your own custom views. For more information, see Collect Data with the Tableau Server Repository for details.

Regenerating the password and certificate

If you need to generate a new password and certificate for internal use, you can use the tabadmin regenerate_internal_tokens command. For example, if you believe
your installation of Tableau Server has been compromised, you should run the `regenerate_internal_tokens` command to generate a new password and SSL certificate.

**Note:** The SSL certificate is used for internal communication between server components and the PostgreSQL database and is independent of any SSL certificate that you might have on the server to use for HTTPS communication between the server and clients that connect to Tableau Server.

To manually regenerate a password and SSL certificate:

1. On the Tableau Server computer, open a command prompt as an administrator and navigate to `<install directory>\Program Files\Tableau\Tableau Server\9.0\bin`.

2. Enter the following:

   ```
   tabadmin stop
   tabadmin regenerate_internal_tokens
   tabadmin config
   tabadmin start
   ```

   See `regenerate_internal_tokens` for more information, including optional switches to specify regeneration of password or certificate.

**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.

Beginning with Tableau Server version 10.2, 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.

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. See assetkeys for more information.
Who has access to the master key?

In a default installation, the Java key store for Tableau Server is installed in `\ProgramData\Tableau\Tableau Server\config\tabsvc\keystores\` folder. If you have installed Tableau on a non-system drive, then the path is `<install drive>:\Tableau\Tableau Server\config\tabsvc\keystores\`. By default, the following accounts have access to this directory:

- Run As User account (if configured)
- NetworkService predefined local Windows account
- LocalSystem predefined local Windows account
- Members of the computer Administrators group

Back up and restore

When you back up Tableau Server using the `tabadmin backup` command, all secrets, keys, and configuration data are written to a secrets storage configuration bundle in the backup file. Since the backup file stores sensitive data, you should store the file in a secure location. You can also set the backup process with the `--no-config` option. With the `--no-config` option set, only content is backed up. All of the configuration data, which includes the secrets storage configuration bundle, is not included in the backup file. See `tabadmin backup` for more information.

When you run restore from a back up file that does not include configuration data, then the restore process will use the existing configuration data on the server where you are running restore. Alternatively, if you restore from a backup that includes the configuration data, then the restore process will use keys and encrypted configuration data from the backup.

You can also run the restore process with the `--no-config` option on the `tabadmin restore` command. In this case, any configuration data in the back up file is ignored and the restore process will use the existing configuration data on the server where you are running restore.
Worker cluster nodes

Adding cluster nodes is handled by a similar process as backup. During the cluster node provisioning process, Tableau Server copies the configuration bundle including encryption keys to each node. Tableau Server distributes the bundle to new cluster nodes.

Upgrade behavior

The Tableau Server Setup program has been updated to handle secure storage.

Upgrade behavior depends upon the version of Tableau Server you are currently running:

- Versions after 10.2: During the upgrade process from a version of Tableau Server where secure storage is already configured, Setup will not make changes to the secure storage configuration.
- Versions prior to 10.2: If you are upgrading from a version of Tableau Server that does not support the implementation of secrets, then upgrade will behave as a normal installation. This means that Setup will generate the master and configuration encryption keys and then encrypt the existing secret values that are stored in the configuration files. In a distributed installation, Setup will then distribute the secrets storage configuration bundle to the nodes in the cluster.

Secrets storage event logging

The following events related to secrets storage are logged in tabadmin.log:

- 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 Server Log File Locations.

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.

As noted in the introduction, the first category of secrets (Run As User and SMTP credentials used by Tableau Server), are those that are generated by server administrators. When you update these passwords with the Tableau Server Configuration Utility, the stored secrets will be automatically encrypted and stored.

The second category of secrets are those that are generated automatically by the various server processes. These secrets are created during installation.

To update automatically generated secrets, run the tabadmin regenerate_internal_tokens command using the --roll_key option. Running this command updates the secrets for each process pair on the server. For the list of secrets and process pairs see the table below in the following section.

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 tabadmin command that will retrieve and decrypt the password for you.

The account you are using to run the tabadmin command must be a member of the Administrators group.

To retrieve a password, open Command Prompt and issue a tabadmin 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:
tabadmin get pgsql.readonly_password

The command will return the password in clear text:

$ tabadmin 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>filestore.zookeeper.password</td>
<td>Password for filestore to connect to zookeeper.</td>
</tr>
<tr>
<td>jdbc.password</td>
<td>Password for the rails Postgres user.</td>
</tr>
<tr>
<td>oauth.google.client_secret</td>
<td>Client secret of the Google Cloud Platform account. See Set up OAuth for Google.</td>
</tr>
<tr>
<td>oauth.quickbooks.consumer_secret</td>
<td>Consumer secret of the Intuit developer account. See Set up OAuth for QuickBooks Online.</td>
</tr>
<tr>
<td>pgsql.adminpassword</td>
<td>Password for the tblwgadmin Postgres user.</td>
</tr>
</tbody>
</table>

**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
<table>
<thead>
<tr>
<th>Configuration Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>pgsql.readonly_password</code></td>
<td>Password for the readonly Postgres user.</td>
</tr>
<tr>
<td><code>pgsql.remote_password</code></td>
<td>Password for the tableau Postgres user.</td>
</tr>
<tr>
<td><code>redis.password</code></td>
<td>Password for Redis.</td>
</tr>
<tr>
<td></td>
<td><strong>Note</strong>: Although the configuration parameter is encrypted in Tableau's configuration files (tabsvc.yml, workgroup.yml), the configuration will still be in plain text in the redis.conf file that is consumed by the Redis application. Redis does not support encrypted/secured passwords.</td>
</tr>
<tr>
<td><code>servercrashupload.proxy_server_password</code></td>
<td>Password for custom proxy server used to upload crash reports.</td>
</tr>
<tr>
<td><code>service.runas.password</code></td>
<td>Password of the Run As users. Stored temporarily.</td>
</tr>
<tr>
<td><code>ssl.key.passphrase</code></td>
<td>Optional passphrase used to protect the Apache SSL key.</td>
</tr>
<tr>
<td><code>svcmonitor.notification.smtp.password</code></td>
<td>SMTP Server password supplied by the administrator through TabConfig.exe.</td>
</tr>
<tr>
<td><code>tabadmins.service.password</code></td>
<td>Password for the service that allows server admins to download log files through the web interface.</td>
</tr>
<tr>
<td>Environment Variable</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------</td>
<td>-------------</td>
</tr>
<tr>
<td>vizportal.openid.client_secret</td>
<td>This is the password (&quot;provider client secret&quot;) used for OpenID Connect SSO. See Configure Tableau Server for OpenID Connect.</td>
</tr>
<tr>
<td>vizsqlserver.external_proxy_password</td>
<td>Password used to authenticate to an external proxy.</td>
</tr>
<tr>
<td>vizsqlserver.extsvc.password</td>
<td>Password for the service that supports R functionality in workbooks.</td>
</tr>
<tr>
<td>wgserver.domain.password</td>
<td>Password used to bind to Active Directory.</td>
</tr>
</tbody>
</table>

### 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.

### 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 External SSL.

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 clickjack 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.

**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 server components. By default, SSL is disabled for communications between server components and the repository.

For more information, see Configure Internal SSL.

**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 contain the hash symbol (#) after the server name (for example, http://myserver#/views/Sales/CommissionModel?:embed=y) are blocked when clickjack protection is enabled on Tableau Server.

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 `tabadmin` commands:

1. `tabadmin stop`
2. `tabadmin set wgserver.clickjack_defense.enabled false`
3. `tabadmin config`
4. `tabadmin start`
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 tabadmin configuration options. If you are running Tableau Server in a distributed deployment, run these commands on the initial node in the cluster. The configuration will be applied across the cluster after you restart the server.

Step 1: Set default directives

Tableau Server includes the set of default directives in the table below.

To set a directives, open tabadmin and follow these steps:

1. Stop Tableau Server:

   `tabadmin stop`

2. Set directives.

   Use the following tabadmin syntax to set the values for a given directive:

   `tabadmin set content_security_policy.directive.<directive_name> <value>`

   For example, to set the `connect_src` directive, run the following command:
```bash
tabadmin set content_security_policy.directive.connect_src
"http://example.com"
```

<table>
<thead>
<tr>
<th>Option</th>
<th>Default value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>content_security_policy.directive.default_src</code></td>
<td>'none'</td>
<td>Serves as a fallback for the other fetch directives.</td>
</tr>
<tr>
<td><code>content_security_policy.directive.connect_src</code></td>
<td>*</td>
<td>Restricts the URLs which can be loaded using script interfaces.</td>
</tr>
<tr>
<td><code>content_security_policy.directive.script_src</code></td>
<td>*</td>
<td>Specifies valid sources for JavaScript.</td>
</tr>
<tr>
<td><code>content_security_policy.directive.style_src</code></td>
<td>* ‘unsafe-inline’</td>
<td>Specifies valid sources for stylesheets.</td>
</tr>
<tr>
<td><code>content_security_policy.directive.img_src</code></td>
<td>* data:</td>
<td>Specifies valid sources of images and favicons.</td>
</tr>
<tr>
<td><code>content_security_policy.directive.font_src</code></td>
<td>* data:</td>
<td>Specifies valid sources for</td>
</tr>
<tr>
<td>Directive</td>
<td>Value</td>
<td>Description</td>
</tr>
<tr>
<td>-----------</td>
<td>-------</td>
<td>-------------</td>
</tr>
<tr>
<td><code>src</code></td>
<td>fonts loaded using <code>@font-face</code>.</td>
<td>Valid values for <code>font-src</code>.</td>
</tr>
<tr>
<td><code>content_security_policy.-directive.frame_src</code></td>
<td><code>* data:</code></td>
<td>Specifies valid sources for nested browsing contexts loading using elements such as <code>&lt;frame&gt;</code> and <code>&lt;iframe&gt;</code>. Valid values for <code>frame-src</code>.</td>
</tr>
<tr>
<td><code>content_security_policy.-directive.object_src</code></td>
<td><code>data:</code></td>
<td>Specifies valid sources for the <code>&lt;object&gt;</code>, <code>&lt;embed&gt;</code>, and <code>&lt;applet&gt;</code> elements. Valid values for <code>object-src</code>.</td>
</tr>
<tr>
<td><code>content_security_policy.-directive.report_uri</code></td>
<td><code>/vizql/csp-report</code></td>
<td>Instructs the user agent to report 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 <code>report-uri</code>.</td>
</tr>
</tbody>
</table>
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. The syntax is as follows:

```
tabadmin set content_security_policy.directive.<new_directive_name> <value>
```

For example, to add the `worker-src` directive, run the following command:

```
tabadmin set content_security_policy.directive.worker-src "http://*.example.com"
```

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. The syntax is as follows:

```
tabadmin set content_security_policy.report_only_directive.<directive_name> <value>
```

For example, to report only on the `script_src` directive, run the following command:

```
tabadmin set content_security_policy.report_only_directive.script_src "http://*.example.com"
```
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:

`tabadmin set content_security_policy.enforce_enabled true`

Step 5: Save changes and start the server.

1. When you are finished setting directives, run the following command:

   `tabadmin config`

2. Restart the server:

   `tabadmin start`
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.
Configure Mutual SSL Authentication for Tableau Server on Windows

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 refuses the connection.

Mutual SSL authentication is not supported on Tableau Mobile.

To configure Tableau Server for mutual SSL, you need the following:

- **SSL certificate file**—A valid PEM-encoded x509 certificate with the extension .crt.
- **SSL certificate key file**—A valid RSA or DSA private key file (with the extension .key by convention).
- **SSL certificate chain file (Optional for Tableau Server, required for Tableau Desktop on the Mac)**—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).

The certificate files should be in the `C:\Program Files\Tableau\Tableau Server-\SSL folder`.  

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.
1 Use SSL for server communication

To configure Tableau Server to use SSL for external communication between Tableau Server and web clients, run the Tableau Server Configuration utility after you have installed Tableau Server. Click the SSL tab, and then select **Use SSL for server communication**.

![Tableau Server Configuration](image)

Specify values for **SSL certificate file** and **SSL certificate key file**.

If the certificate key file requires a passphrase enter it in the field, **SSL certificate key passphrase**. (The passphrase you enter will be encrypted while at rest). Alternatively, you can provide a path to a key file that is not passphrase protected.

2 Use mutual SSL

To add mutual authentication between the server and each client and allow for automatic sign-in experience, select **Use mutual SSL and automatic login with client certificates**.
Specify the **SSL CA certificate file**. The SSL CA certificate file identifies the certificate of the Certificate Authority (for example, Verisign). For information on how to configure multiple Certificate Authorities, see Configure External SSL.

Click OK to close the Tableau Server Configuration utility, and then start Tableau Server.

Additional options for mutual SSL

**Fallback authentication**

When Tableau Server is configured for mutual SSL, authentication is automatic and a client must have a valid certificate. If you need a fallback option, use the `tabadmin set ssl.client_certificate_login.fallback_to_password true` command to configure Tableau Server to allow user name / password authentication. Setting this option to **true** allows Tableau Server to fall back to using user name and password for authentication if SSL certificate authentication fails.
Username mapping

When Tableau Server is configured for mutual SSL, the server gets the user name from the client certificate so the client can be automatically signed in. 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.

You can override either of these defaults to set Tableau Server to use the CN (Common Name) by using the `tabadmin set ssl.client_certificate_login.mapping_strategy` command.

Certificate Revocation List (CRL)

You may 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. To specify a CRL, use the `tabadmin set ssl.revocation.file` command. For more information, see `tabadmin set Commands`.

Configure External SSL

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 web browser and the server or Tableau Desktop and the server is protected. Steps on how to configure the server for SSL are described in the topic below; however, you must first acquire a certificate from a trusted authority, and then import the certificate files into Tableau Server.

Mutual SSL authentication is not supported on Tableau Mobile.
Looking for Tableau Server on Linux? See Configure SSL for External HTTP Traffic to and from Tableau Server (Linux).

If you are running a Tableau Server cluster and you want to use SSL, see Configure SSL for a Cluster, below, for recommendations.

1. 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. Wildcard certificates, which allow you to use SSL with many host names within the same domain, are also supported.

   **Note:** Be sure to use a SHA-2 (256 or 512 bit) certificate. All major browsers will display warnings when connecting to SHA-1 certificates. By the end of 2017, it's likely that most browsers will no longer connect to servers that are presenting SHA-1 certificates.

   Some browsers will require additional configuration to accept certificates from certain providers. Refer to the documentation provided by your certificate authority.

2. Place the certificate files in a folder named SSL, parallel to the Tableau Server 10.5 folder. For example:

   C: \ Program Files \ Tableau \ Tableau Server \ SSL

   This location gives the account that's running Tableau Server the necessary permissions for the files. You may need to create this folder.

3. Stop Tableau Server. From the Start menu, click All Programs > Tableau Server 10.5 > Stop Tableau Server.

4. Open the Tableau Server Configuration Utility by selecting Start > All Programs > Tableau Server 10.5 > Configure Tableau Server on the Start menu.
5. In the Configuration Tableau Server dialog box, select the **SSL** tab.

![Configuration Tableau Server dialog box](image)

6. Select **Use SSL for server communication** and provide the location for each of the following certificate files:

- **SSL certificate file**—Must be a valid PEM-encoded x509 certificate with the extension `.crt`.

- **SSL certificate key file**—Must be a valid RSA or DSA private key file (with the extension `.key` by convention). If the certificate key file requires a passphrase enter it in the field, **SSL certificate key passphrase**. (The passphrase you enter will be encrypted while at rest). Alternatively, you can provide a path to a key file that is not passphrase protected.

  **Note:** If you create a certificate key file with a passphrase, you cannot reuse the SSL certificate key for SAML.

- **SSL certificate chain file** (Optional for Tableau Server, required Tableau Desktop on the Mac)—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).

7. (optional) If you are using SSL for server communication and want to configure SSL communication between Tableau Server and clients using certificates on both the server and clients:

   • Select **Use mutual SSL and automatic login with client certificates**.

   **Note:** Tableau Server does not support mutual SSL and SAML together.

   • In **SSL CA certificate file**, browse to the location for the certificate file. The SSL CA certificate file must be a valid PEM-encoded x509 certificate with the extension .crt.

   **Note:** If you have multiple trusted Certificate Authorities (CAs) you can copy and paste the entire contents of each CA certificate, including the "BEGIN CERTIFICATE" and "END CERTIFICATE" lines, into a new file, then save the file as CAs.crt. In **SSL CA certificate file**, browse to the location of this new file.

8. Click **OK**.

   If the Run As User account on the Tableau Server is not set to the default Network Services user, then you will be prompted for the Run As User password.

9. Start Tableau Server again. From the Start menu, click **All Programs > Tableau Server 10.5 > Start Tableau Server**.

**Configure SSL for a Cluster**

You can configure a Tableau Server cluster to use SSL. If the primary 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 earlier.

**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. Place the same SSL certificate and key file that you used for the initial node on each subsequent Tableau Server node that runs a gateway process. Use the same folder
location on all computers.

5. If you are using mutual SSL, place the SSL CA certificate file in the same location on all computers that run a gateway process.

You do not need to do any additional configuration on the subsequent nodes.

Example

Say you have a cluster that includes a primary Tableau Server node and three worker nodes, with gateway processes running on the primary, Worker 2 and Worker 3. In this situation, you configure the primary Tableau Server for SSL, and then copy the same SSL certificate and key files to Worker 2 and Worker 3, to the same location as on the primary.

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.

**Note:** Tableau Server only supports 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 install directory at the following location. Use this log to troubleshoot validation and encryption issues:

```
C:\ProgramData\Tableau\Tableau Server-data\tabsvc\logs\httpd\error.log
```

Configure Internal SSL

This article applies to Tableau Server on Windows. For Tableau Server on Linux, see Configure SSL for Internal Postgres Communication.
You can configure Tableau Server to use Secure Sockets Layer (SSL) for encrypted communications on all traffic between the Postgres repository and other server components. By default, SSL is disabled for communications between server components and the repository.

1. Open the Tableau Server Configuration Utility by selecting Start > All Programs > Tableau Server 10.5 > Configure Tableau Server.

2. In the Tableau Server Configuration dialog box, click the SSL tab.

3. Select one of the following options:

   - **Required for all connections**
     
     When this option is selected, Tableau Server uses SSL for communications between the repository database and other server components. In addition, direct connections to Tableau Server (connections using the "tableau" or "readonly" users) must use SSL.

   - **Optional for direct user connections**
     
     This option configures Tableau Server to use SSL between the repository and other server components and supports but does not require SSL for direct connections by "tableau" or "readonly" users.

   - **Off for all connections** (the default)
     
     This option disables SSL for internal communications and direct connections.

4. Click OK.

For more information on 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, and the REST API.

Before you can require SSL for direct connections, you need to enable Postgres (repository) SSL. For information, see Configure Internal SSL. If you know it’s enabled, complete the following steps:

1. Run the following command:

   ```
tabadmin regenerate_internal_tokens --certs
   ```

   This generates the server’s SSL certificate files.

2. Locate the certificate file.

   You can find the location in the workgroup.yml file. This file is located on the primary Tableau Server node, in the `\ProgramData\Tableau\Tableau Server-data\tabsvc\config` folder.

   The location of the SSL certificate and key files are listed in the file. For example:

   ```
   postgres.ssl.cert.file: C:/ProgramData/Tableau/Tableau Server-data/tabsvc/config/postsql/server.crt
   postgres.ssl.key.file: C:/ProgramData/Tableau/Tableau Server-data/tabsvc/config/postsql/server.key
   ```

3. On the computer that will make the direct connections, do the following:

   a. Copy **server.crt** to the computer.

   ```
   **Note:** Do not copy **server.key** to the client computer. This file should reside only on the server.
   ```

   b. Import the certificate into the computer’s certificate store.
For information, use the documentation from the operating system manufacturer.

See also

`tabadmin regenerate_internal_tokens`

`tabadmin`

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.

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

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:

```plaintext
username, domain\username, or username@domain.
```

**For example:** asmith, example.org\asmith, or asmith@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 following command to map a client certificate to a user identity in Tableau Server:

```bash
ssl.client_certificate_login.mapping_strategy <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:

```bash
tabadmin stop

tabadmin set ssl.client_certificate_login.mapping_strategy cn

tabadmin configure

tabadmin start
```
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, asmith@example.org and asmith@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 asmith.

To avoid incorrect user-name mapping, make sure the client certificates include fully qualified user names with the domain, using the format asmith@example.org or example.org/asmith.

Troubleshooting Mutual SSL Authentication

This topic describes possible mutual (two-way) SSL authentication issues and their causes, the messages that users might see, and possible mitigation for the issues.
Issues

We couldn’t find a valid client certificate. Contact your Tableau Server administrator.

Invalid user name or password

We couldn’t find your user name in the client certificate. Contact your Tableau Server administrator or sign in using your Tableau Server account.

Users unexpectedly see a sign-in dialog box that displays an error message

We couldn’t find your user name in the client certificate. Contact your Tableau Server administrator Certificate does not contain a valid Tableau Server user name.

The user is signed in using an unexpected user name (LDAP mapping)

The user is signed in as the incorrect user (UPN or CN mapping)

For more information about mutual SSL authentication and LDAP, UPN, and CN user mapping, see the following topics:

- Configure Mutual SSL Authentication for Tableau Server on Windows
- Mapping a Client Certificate to a User During Mutual Authentication

We couldn't find a valid client certificate. Contact your Tableau Server administrator.

The client is missing a certificate.
If the client has no client certificate, the user sees this message during authentication:

We couldn't find a valid client certificate. Contact your Tableau Server administrator.

To resolve the issue, the user should contact the system administrator to generate a certificate for the client computer.

Invalid user name or password

The client doesn't support mutual SSL authentication.

Versions of Tableau Desktop older than version 9.1 do not support mutual SSL authentication. If an older version of Tableau Desktop is used to connect to Tableau Server that is configured for mutual SSL authentication, the following can occur:

- If Tableau Server is configured to use fallback authentication, the client displays a sign-in dialog box and the user can enter a user name and password.

- If the server is not configured to use fallback authentication, the user sees the following message and cannot connect to the server:

  Invalid user name or password

For more information about fallback authentication, see Configure Mutual SSL Authentication for Tableau Server on Windows.

We couldn't find your user name in the client certificate. Contact your Tableau Server administrator or sign in using your Tableau Server account.

Client certificates are not published to Active Directory.
If Tableau Server is configured to use Active Directory for authentication, and if user mapping is set to LDAP, Tableau Server sends the client certificate to Active Directory for authentication. However, if client certificates have not been published to Active Directory, authentication fails and the user sees the following message:

We couldn't find your user name in the client certificate. Contact your Tableau Server administrator or sign in using your Tableau Server account.

To resolve this issue, the system administrator should make sure that client certificates are published to Active Directory. Alternatively, the server should be configured to use a different user mapping (UPN or CN), and the system administrator should be sure that client certificates contain user names in the UPN or CN fields.

**Users unexpectedly see a sign-in dialog box that displays an error message**

If Tableau Server is configured to use mutual SSL authentication and certificates are available for use with users’ computers, a user should not see a sign-in dialog box, because Tableau Server uses the certificate to authenticate the user. However, if the server does not recognize the user name in the certificate, the user sees a sign-in dialog box with an error message that indicates why the certificate was not used. This can occur when all of the following conditions are true:

- Fallback authentication is enabled.

- If the server is using UPN or CN mapping, the user name in the certificate’s UPN or CN field is not recognized. If the server is using LDAP mapping, the certificate is not mapped to the user in Active Directory.

To resolve this issue, the system administrator should do the following, depending on how user mapping is configured on Tableau Server:
- LDAP mapping: Make sure that the certificate is linked to the user, that the certificate is available for use with the user's computer, and that the user is configured as a Tableau Server user.

- UPN or CN mapping: Make sure that the certificate is available for the user's computer, that the user name is in the certificate's UPN or CN field, and that the user name matches the user name on Tableau Server (including domain).

We couldn't find your user name in the client certificate. Contact your Tableau Server administrator

Certificate does not contain a valid Tableau Server user name.

The user name in the UPN or CN fields is missing or invalid

When Tableau Server is configured to use UPN or CN mapping, the server reads the user's name from the UPN or CN field of the certificate and then looks up the user name in Active Directory or in the local repository on Tableau Server. (The specific field that the server reads depends on which mapping—UPN or CN—the server is configured to use.) If the field that is supposed to contain the user name has nothing in it, the user sees the following message:

We couldn't find your user name in the client certificate. Contact your Tableau Server administrator.

If a client certificate contains a user name but Active Directory and Tableau Server don't recognize the user name, the user sees the following message:

Certificate does not contain a valid Tableau Server user name.

This can occur when all of the following conditions are true:
Tableau Server is configured to use UPN or CN mapping.

Fallback authentication is not enabled.

The client certificate has no user name in the UPN or CN field, or the user name in the UPN or CN field does not match a user name in Active Directory or on Tableau Server.

To resolve this issue, the system administrator should make sure that the user’s certificate has the correct user name in the UPN or CN fields of the certificate.

The user is signed in using an unexpected user name (LDAP mapping)

When the server is configured to use Active Directory authentication and LDAP mapping, the certificate is linked to a user in Active Directory. If the certificate contains a user name in the UPN or CN field, that user name is ignored.

If the intention is that the user should be signed in with the user name in the UPN or CN fields, the server should be configured to use UPN or CN mapping.

The user is signed in as the incorrect user (UPN or CN mapping)

Under some circumstances, the user name in a UPN or CN field in the client certificate can be ambiguous. The result is that a user is signed in to the incorrect identity.

For more information about the conditions under which this issue can occur, see the section “Address user-name ambiguity in multi-domain organizations” in Mapping a Client Certificate to a User During Mutual Authentication.
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 Linux? 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. Set the OpenSSL configuration environment variable (optional).
2. Generate a key file.
3. Create a Certificate Signing Request (CSR).
4. Send the CSR to a certificate authority (CA) to obtain an SSL certificate.
5. 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.

Set the OpenSSL configuration environment variable (optional)

To avoid using the `-config` argument with every use of openssl.exe, you can use the `OPENSSL_CONF` environment variable to ensure that the correct configuration file is used and all configuration changes made in subsequent procedures in this article produce expected results (for example, you must set the environment variable to add a SAN to your certificate).

Open the Command Prompt as an administrator, and run the following command:

```
set OPENSSL_CONF=c:\Program Files\Tableau\Tableau Server-<version>\apache\conf\openssl.cnf
```

Notes:

- When setting the Open SSL configuration environment variable, do not enclose the file path with quotation marks.

- If you are using a 32-bit version of Tableau Server on a 64-bit computer, run the `set OPENSSL_CONF=c:\Program Files (x86)\Tableau\Tableau Server-<version>\apache\conf\openssl.cnf` command instead.

Generate a key

Generate a key file that you will use to generate a certificate signing request.

1. Open the Command Prompt as an administrator, and navigate to the Apache directory for Tableau Server. For example, run the following command:

   `cd C:\Program Files\Tableau\Tableau Server\<version>\apache`

2. Run the following command to create the key file:
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 certificate.

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:

   openssl.exe req -new -key yourcertname.key -out yourcertname.csr

   If you did not set the OpenSSL configuration environment variable, OPENSSL_CONF, you might see either of the following messages:

   - An error message about the config information being unable to load. In this case, retype the command above with the following parameter: -config ..\conf\openssl.cnf.

   - A warning that the /usr/local/ssl directory cannot be found. This dir-
The file is created successfully.

To set an OpenSSL configuration environment variable, see Set the OpenSSL configuration environment variable (optional) section in this article.

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. Open Windows Explorer and browse to the Apache conf folder for Tableau Server.
   
   *For example: C:\Program Files\Tableau\Tableau Server-<version>\apache\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.

   ```
   UTF8Strings
   # so use this option with caution!
   string_mask = nomapstr

   # 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.

[ v3_req ]
# Extensions to add to a certificate request
basicConstraints = CA:FALSE
keyUsage = nonRepudiation, digitalSignature, keyEncipherment
subjectAltName = @alt_names

[alt_names]
DNS.1 = domain1
dNS.2 = domain2
dNS.3 = etc

5. Save and close the file.

6. Complete the steps in Create a certificate signing request to send to a certificate authority section, above.

Additional information

If you prefer to use a different version of OpenSSL, you can download it from Open SSL for Windows.
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. For more information see Upgrade Tableau Server Overview.

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.

Specially, 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 tabadmin 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).

```
tabadmin stop

tabadmin set ssl.protocols "all -SSLv2 -SSLv3 -TLSv1 -TLSv1.1"
tabadmin configure
tabadmin start
```

4. Disable Triple-DES cipher suite

The Triple-DES cipher suite is no longer considered adequate to encrypt sessions on the internet. Specifically, running Triple-DES ciphers leaves the Tableau Server vulnerable to information disclosure and denial of service attacks. You can learn more at the National Vulnerability Database webpage for CVE-2016-2183.

Triple-DES is enabled by default on the version of OpenSSL that is running on Tableau Server. However, other deprecated cipher suites (MD5 and RC4) are disabled. To add Triple-DES to the list of disabled ciphers, run the following commands. (The !aNULL parameter forces clients to use a legitimate cipher.)

```
tabadmin stop

tabadmin set ssl.ciphersuite HIGH:MEDIUM:!aNULL:!MD5:!RC4:!3DES
tabadmin configure
tabadmin start
```
5. 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 Internal SSL.

6. Enable firewall protection

Tableau Server was designed to operate inside a protected internal network. Do not set up Tableau Server in the same network with your internet gateway or in a DMZ. Tableau Server must be protected by external firewall. The platform firewall, such as the Windows firewall, should be enabled 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. By default, the Tableau installation process configures ports in the Windows firewall so that server components can communicate with each other. If you’re configuring a different firewall, or if you need to configure the Windows firewall after you’ve installed Tableau Server, see Tableau Server Ports to understand which ports and services Tableau Server requires.

To prevent a passive attacker from observing communications between nodes, configure a segregated virtual LAN or other network layer security solution.

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 Configure a reverse proxy server.
7. 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 C:\ProgramData\Tableau directory. By default, the permissions on these directories are restrictive, therefore we do not recommend changing permissions at the directory level.

8. Update the Tableau Server Run As User account

By default, Tableau Server runs under the predefined Network Services (NT Authority\Network Service) Windows account. Using the default account is acceptable in scenarios where Tableau Server does not need to connect to external data sources that require Windows authentication. However, if your users require access to data sources that are authenticated by Active Directory, update the Run As User to a domain account. It’s important to minimize the rights of the account that you use for the Run As User. For more information, see Run As User.

9. Generate fresh asset keys

Tableau Server encrypts embedded database credentials before they are stored in the repository. The credentials are encrypted with asset keys. We recommend that after you install Tableau Server, you generate new encryption keys for your deployment. To do this, use the tabadmin assetkeys command.

10. Refresh server token and encryption key

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. In addition, the encryption key that internal SSL uses to encrypt traffic to Postgres repository is also generated at during Setup. If your organization follows a security policy to update shared secrets and encryption keys on a regular schedule, you should include the
token and key in that process. See the tabadmin regenerate_internal_tokens command for more information.

11. 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 sequence of tabadmin commands:

```
tabadmin stop

tabadmin set api.server.enabled false

tabadmin configure

tabadmin start
```

You can disable REST API only on versions of Tableau Server 9.3 and later.

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 sequence of tabadmin commands:

```
tabadmin stop

tabadmin set service.jmx_enabled false

tabadmin configure

tabadmin start
```
12. 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. Before you do that, you must enable session lifetime timeout. Use the following sequence of `tabadmin` commands.

```
tabadmin stop

tabadmin set wgserver.session.apply_lifetime_limit true

tabadmin set wgserver.session.lifetime_limit "value", where value is the number of minutes. The default is 1440, which is 24 hours.

tabadmin set wgserver.session.idle_limit "value", where value is the number of minutes. The default is 240.

tabadmin configure

tabadmin start
```

13. 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 Run As User account.

To prevent unwanted access to files, we recommend that you configure safelist (sometimes referred to as "whitelist") functionality. This lets you limit Run As User 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 `tabadmin` commands:

   `tabadmin stop`

   `tabadmin set native_api.allowed_paths "path", where path is the directory to add to the safelist. Note! 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:

   `tabadmin set native_api.allowed_paths "c:\data-sources;c:\HR\data"

   `tabadmin configure`

   `tabadmin start`

14. 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.

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 `tabadmin` commands on Tableau Server:

   `tabadmin stop`

   `tabadmin set gateway.http.hsts: 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 `tabadmin set gateway.http.hsts_options: max-age=<seconds>`. For example, to set HSTS policy time period to 30 days, enter `tabadmin set gateway.http.hsts_options: max-age=2592000`.

`tabadmin configure`

`tabadmin start`

15. 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.

Change List

<table>
<thead>
<tr>
<th>Date</th>
<th>Change</th>
</tr>
</thead>
<tbody>
<tr>
<td>December 2016</td>
<td>Added &quot;Disable Triple-DES cipher suite&quot;</td>
</tr>
<tr>
<td>May 2017</td>
<td>Added &quot;Enable HTTP Strict Transport Security for web browser clients;&quot; &quot;Disable Guest Access&quot;</td>
</tr>
</tbody>
</table>

Manage Licenses

Licensing Overview

Tableau Server licenses have two aspects: the license model (term or perpetual) and the license metric (user-based or core-based). Term licenses are user-based, and perpetual licenses usually core-based.

License model: term or perpetual

Tableau Server can be licensed under two models: term licensing and a perpetual licensing. Term licenses, also known as subscription licenses, allow you to use and update Tableau Server for a specified period of time.

Perpetual licenses do not expire, so you can continue to use Tableau Server as long as you want. However, to get access to product updates and technical support you must purchase Support and Maintenance services.
License metric: user-based or core-based

In addition to the license model, your license is also 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 addi-
tional information on licensing and management tasks related to Tableau Desktop, see The Tableau Desktop Deployment Guide.

View Server Licenses

Server administrators can view the license and product key information for Tableau Server.

Tableau Server site roles do not correspond to user licenses that you purchase from Tableau (if you are using user-based licensing instead of core-based server licensing). Those licenses allow a certain number of users on the server.

To view server licenses

You can view licenses from the server web pages, on the Licenses page, or from the command line, using the `tabadmin licenses` command.

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.

If you have a user-based Tableau Server license, you can review how these levels have been distributed.

If you have a core-based Tableau Server license, the Licenses page shows how many cores are allowed, how many have been licensed, and how many are in use (and on what server computers).

Also see:

- Licensing Overview
- Handle an Unlicensed Server.

Refresh Maintenance Date for the Product Key

If the maintenance date for your product key is not up-to-date, refresh the product key. Refreshing the product key will update the product key with the date that your current maintenance period expires and stops expiration messages you may have already received. 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.

Refresh the product key

1. On the Tableau Server computer, click Start and select **All Programs > Tableau Server**.
2. Right-click **Manage Product Keys** and select **Run as administrator**.
3. In **Manage Product Keys**, select the product key, and then click **Refresh**.

![Manage Product Keys](image)

The latest maintenance date will be retrieved and updated.

4. Verify that the **Maintenance Expires** date has been updated.
5. Click **Close**.
6. Restart the server processes if you are refreshing a user-based license maintenance date. If you are refreshing a core-based license, then you do not need to restart the server.

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.

1. Start the Product Key Manager:
   
   In Windows, select **Start > All Programs > Tableau Server <version> > Manage Product Keys**.

2. Click **Activate** in the Manage Product Key dialog box:

3. Enter or paste your new product key and click **Activate**:
4. Restart Tableau Server after registration is complete.

Automate Licensing Tasks

Several command line options allow you to perform common licensing tasks for Tableau Server using automated scripts.

If you want to perform these licensing tasks for Tableau Desktop using automated scripts, see Deploy Tableau Desktop in the Desktop Deployment Guide.

Command Line Tool Options and Better Script Automation

You can use a utility called `tabinstallck.exe` to script the activation, refresh, and deactivation of server license keys. `tabinstallck.exe` is installed in the Tableau Server `\bin` directory on the primary server node. For example:

`D:\Tableau\Tableau Server\<version>\bin`
Note: You can also use `tabadmin` to perform these licensing tasks, but in order to do this Tableau Server must already be deployed and configured, so `tabinstallck.exe` provides a way to script the licensing step for a new Tableau Server installation that is not completely configured.

To use `tabinstallck.exe`, you need to run the commands or script as an administrator on the primary Tableau Server node.

Activate

Activate a Tableau Server license using the `-activate` option:

`start/wait tabinstallck.exe -activate <product_key>`

Refresh

Refresh a Tableau Server license using the `-refresh` option:

`/wait tabinstallck.exe -refresh <product_key>`

Deactivate

Deactivate a Tableau Server license using the `-return` option:

`start/wait tabinstallck.exe -return <product_key>`

Example

An example of a script to activate Tableau Server might look like this:

```bash
@echo off
start/wait tabinstallck.exe -activate <product_key>
if %errorlevel% EQU 0 (ECHO SUCCESS) ELSE (ECHO FAILED)
```
Error Handling

When you use `tabinstallck.exe` to automate Tableau Server licensing tasks, status information is returned as an exit code in the ERRORLEVEL environment variable. If activation was successful, the exit code is 0 (zero). The table below lists exit codes from the activation process and what they mean. You can also find error information in the `tabinstallck.log` file. This log file is located in the Tableau Server \logs folder, by default:

```
C:\ProgramData\Tableau\Tableau Server\logs
```

The ProgramData folder is hidden by default.

This example script shows how to activate Tableau Server and check for specific errors, such as an invalid product key (536871018) and general licensing issues (536871012).

```batch
@echo off
start/wait tabinstallck.exe -activate <product_key>
if %errorlevel% EQU 536871018 (ECHO TABLEAU_ERROR_LICENSING_INVALID_KEY)
if %errorlevel% EQU 536871012 (ECHO TABLEAU_ERROR_LICENCING_GENERAL)
if %errorlevel% EQU 0 (ECHO SUCCESS)
```

<table>
<thead>
<tr>
<th>Exit Code</th>
<th>Message</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>536871012</td>
<td>TABLEAU_ERROR_LICENSING_GENERAL</td>
<td>An unknown error occurred during the licensing operation.</td>
</tr>
<tr>
<td>536871013</td>
<td>TABLEAU_ERROR_LICENSING_INTERNAL</td>
<td>The product is unable to start due to an internal licensing error.</td>
</tr>
<tr>
<td>536871014</td>
<td>TABLEAU_ERROR_LICENSING_SETUP</td>
<td>The product is unable to start due to an internal licensing error.</td>
</tr>
<tr>
<td>536871015</td>
<td>TABLEAU_ERROR_LICENSING_CONNECT</td>
<td>An error occurred contacting the Tableau license server.</td>
</tr>
<tr>
<td>Code</td>
<td>License Error Type</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>---------------------------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>536871017</td>
<td>TABLEAU_ERROR_LICENSING_ACTIVATION_ASR</td>
<td>An error occurred while activating the ASR.</td>
</tr>
<tr>
<td>536871016</td>
<td>TABLEAU_ERROR_LICENSING_ACTIVATION_GENERAL</td>
<td>Activation failed, please try again.</td>
</tr>
<tr>
<td>536871018</td>
<td>TABLEAU_ERROR_LICENSING_ACTIVATION_INVALID_KEY</td>
<td>Invalid product key.</td>
</tr>
<tr>
<td>536871019</td>
<td>TABLEAU_ERROR_LICENSING_ACTIVATION_OFFLINE</td>
<td>Unable to complete offline activation.</td>
</tr>
<tr>
<td>536871020</td>
<td>TABLEAU_ERROR_LICENSING_ACTIVATION_INVALID_MODEL</td>
<td>An error was returned from the license server. Check the Tableau log file for more information.</td>
</tr>
<tr>
<td>536871021</td>
<td>TABLEAU_ERROR_LICENSING_ACTIVATION_MAX_REHOSTS</td>
<td>Failed to deactivate the license because you have reached the limit of deactivations for this license.</td>
</tr>
<tr>
<td>536871022</td>
<td>TABLEAU_ERROR_LICENSING_UNLICENSED_GENERAL</td>
<td>An unknown error occurred during the unlicensing operation. Check your product key or contact support.</td>
</tr>
<tr>
<td>536871023</td>
<td>TABLEAU_ERROR_LICENSING_UNLICENSED_NONE_FOUND</td>
<td>No licenses found.</td>
</tr>
<tr>
<td>536871024</td>
<td>TABLEAU_ERROR_LICENSING_UNLICENSED_EXPIRED</td>
<td>Only expired product keys were found.</td>
</tr>
<tr>
<td>536871025</td>
<td>TABLEAU_ERROR_</td>
<td>Your trial for this version has ended.</td>
</tr>
<tr>
<td>Code</td>
<td>Message</td>
<td></td>
</tr>
<tr>
<td>--------------</td>
<td>----------------------------------------------</td>
<td></td>
</tr>
<tr>
<td>536871026</td>
<td>The product key is expired.</td>
<td></td>
</tr>
<tr>
<td>536871027</td>
<td>Maintenance has expired.</td>
<td></td>
</tr>
<tr>
<td>536871028</td>
<td>Your product keys could not be synchronized.</td>
<td></td>
</tr>
<tr>
<td>536871029</td>
<td>An error occurred during product registration.</td>
<td></td>
</tr>
<tr>
<td>536871030</td>
<td>An unknown error occurred during licensing operations.</td>
<td></td>
</tr>
<tr>
<td>536871031</td>
<td>An unknown error occurred during licensing operations.</td>
<td></td>
</tr>
</tbody>
</table>

**Handle an Unlicensed Server**

Tableau offers two licensing models: user-based and core-based. 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 it allows. Each user is assigned a unique user name on the server and is required to identify himself 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. You can see your products keys and add new ones by selecting Start > All Programs > Tableau Server > Manage Product Keys.

Unlicensed Core-Based Server

A core-based server can become unlicensed for a variety of reasons. A common problem is that any Tableau Server node has more cores than the license allows. When the server is unlicensed you may not be able to start or administer the server. You can, however, manage your licenses using the tabadmin command line tool. Follow the steps below to see a list of your licenses and number of cores by machine.

1. Open a command prompt as administrator and navigate to the Tableau Server bin directory. For example, on a default install, type the following: cd C:\Program Files\Tableau\Tableau Server\10.5\bin

2. Type the following: tabadmin licenses -p.

Add Users to the Server

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.
The following procedure describes how to add users to the server. There are two approaches you can take: One at a time (described below) or in batches using the Import command, which relies on a CSV file (described in Import Users and CSV Import File Guidelines).

To add a user to the server

1. In the site menu, click **Manage All Sites**, click **Users**, and then click **Add Users**.

2. If you are using local authentication, click **New User**. If you are using Active Directory, click **Active Directory User**.

   Enter a user name.
• **Local authentication**: If the server is using local authentication, using an email address for the user name is the best way to avoid user name collisions (for example, jsmith@example.com instead of jsmith).

• **Active Directory**: 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 domain will be assumed.

Before adding users, 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.

**Note**: Do not enter the user’s full name in this field; it 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 on site roles, see Set Users’ Site Roles.

![New User form](image)

4. Click **Create**.

**Sign in to Tableau Server Admin Pages**

This topic explains how to sign in to the Tableau Server as an administrator.

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](#).

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 Server Admin Pages

As a server administrator, you can access all of the menus and pages in Tableau Server for server and site management. If your server is configured for multiple sites, the site menu is available for navigation. Click Manage All Sites in the site menu to access server administration pages.

The server administrator pages include server-wide settings that you will use to configure, monitor, and maintain Tableau Server.

Server Administrator Pages

In a single-site deployment, all server and site menus are available to you in the main menu. To create a site, click Settings > Add a Site.

On a multi-site server, when a site is selected, you will see these menus:

To access server administration pages, click the site menu, and then select Manage All Sites:
On a multi-site server, these are the server administration menus. The site menu text changes to **All Sites** to let you know you are managing server-wide settings.

To return to the site administration menus, click **All Sites**, and select the site you want to manage.
Server administrators can:

- 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 site-related settings, you must first navigate to a specific site. Within each site, you can:

- Manage content and assign permissions.
- 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).

Manage Sites

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.

In this article

What is a site
Authentication and sign-in credentials
The Default site
Why or why not add sites
Administrator-level access to sites
Licensing and user limits
See also
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.

Authentication and sign-in credentials

All sites on a server use the same Run As user account. By default they use the same user authentication type. You configure both of these settings when you install Tableau Server. For information, see Configure General Server Options.

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.

**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 two administrator-level site roles: Server Administrator and Site Administrator.

The **Server Administrator** site role allows full access to Tableau Server. 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 the **Site Administrator** site role 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.

By default, the Site Administrator site role allows 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.

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

Server users can be added to multiple sites, and their site roles and permissions set on each site. A user who belongs to several sites does not need a license for each site. Each server user only needs 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; server administrators are excluded. For example, if a site has 90 licensed users, 20 unlicensed users, and one server administrator, the user count is 90.

See also

- 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.

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.
• If you’ve added sites before, in the site menu, click **Manage All Sites**, and then click **New Site**.

• 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 **General** tab.
2. Enter a Site name and Site ID for the site. (If you are editing the Default site, you cannot change the Site ID.)

   ![Sites GUI](image)

<table>
<thead>
<tr>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>Site Name:</td>
</tr>
<tr>
<td>Enter Site Name</td>
</tr>
<tr>
<td>Site ID:</td>
</tr>
<tr>
<td>Enter Site ID</td>
</tr>
<tr>
<td>URL:</td>
</tr>
</tbody>
</table>

**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.

<table>
<thead>
<tr>
<th>Name</th>
<th>Users</th>
<th>Site Admins</th>
<th>Max Users</th>
<th>Storage Used</th>
<th>Max Storage</th>
<th>Status</th>
</tr>
</thead>
<tbody>
<tr>
<td>Default</td>
<td>10</td>
<td>1</td>
<td>Server limit</td>
<td>12.9 MB</td>
<td>Server limit</td>
<td>Active</td>
</tr>
<tr>
<td>MyFirstSite</td>
<td>0</td>
<td>0</td>
<td>Server limit</td>
<td>0 B</td>
<td>Server limit</td>
<td>Active</td>
</tr>
<tr>
<td>MySecondSite</td>
<td>0</td>
<td>0</td>
<td>Server limit</td>
<td>0 B</td>
<td>Server limit</td>
<td>Active</td>
</tr>
</tbody>
</table>

4. Select whether only server administrators can add and remove users and change their site roles, or whether you want to allow site administrators to manage users on this site.
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 Handle an Unlicensed Server.

- **<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. Select **Allow users to use web authoring** if you want to enable browser-based authoring for site content.

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.

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 Tableau user Help.

8. Select **Allow users to subscribe to workbooks and views** to let site users subscribe to views and receive regular emails of them. Select **Allow content owners to subscribe other users to workbooks and views** to let administrators, project leaders, and content owners set up subscriptions for other users. 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 snapshots 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.

**In this article**

**Before you export**
- Site migration options
- What information is or isn't preserved in a site export
- Prepare the source and target sites
- Tips for importing to a target with fewer users or schedules than the source site

**Steps to migrate a site**
1. Export a site
2. Generate the import mapping files
3. Verify that site settings are mapped correctly
4. Import the correctly mapped files 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. However, 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 your representative will 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, site quota, and other site settings.

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 subscription schedules. These schedules currently are not supported to map to Tableau Server.
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.

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, 10.5 through 10.5.x. You cannot import a site that is outside the target site’s version family. For example, 10.4 to 10.5. For version information, see the Tableau Release Notes.

- **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 tabadmin 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 a pair of double quotation marks (" " ) with nothing in between.

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 user authentication
You can export from and import to sites that do not use the same user authentication method, 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 Configure Server Alerts.

• Check schedules
The Schedules page lists the existing schedules for extract refreshes and subscriptions.

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.

![Subscriptions screenshot](image)

**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

1. On the Tableau Server machine, open a command prompt as an administrator and navigate to the server's bin directory. For example:

   C:\Program Files\Tableau\Tableau Server\10.5\bin

2. Type the following command:

   tabadmin exportsite <site ID> --file <filename or path>.

   Tableau Server must be running when you use the exportsite 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 C:sites\exported_sites\export-file.zip, type the following:

   tabadmin exportsite weather-data --file C:\exported_sites\export-file.zip
If you are using Windows PowerShell to run the command, and you want to export the Default site, enclose the double quotation marks that represent the site ID within single quotations. For example:

```
tabadmin exportsite """" --file C:\exported_sites\default.zip
```

For more examples, see exportsite.

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, open a command prompt as an administrator and navigate to the server’s bin directory. For example:

   ```
   C:\Program Files\Tableau\Tableau Server\10.5\bin
   ```

2. Type the following command (Tableau Server must be running):

   ```
   tabadmin importsite <site ID> --file <filename or path>
   ```

   - `<site ID>` is the site ID of the target site.
   - `<filename or path>` is the full path to the .zip file you created when you ran the exportsite command.

   For example, to import the file `C:sites\exported_sites\sales_export.zip` to a site with the site ID `esales`, type the following:

   ```
   tabadmin importsite esales --file C:sites\exported_sites\sales_export.zip
   ```

   For other examples and options, see importsite.

   The importsite command creates the mapping files you will verify in the next section.

   You can find these files in the following location:
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 `import-site` command.
2. Using Microsoft Excel (recommended) or a text editor, open each CSV file you see in the mappings folder.

   Each file shows how items from the source site will be mapped to the target site during the import process.

3. Verify that the mappings are correct. Replace any entry consisting of question marks (???) 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.

4. If you make edits, save your changes and preserve the CSV file formatting. Leave the mapping files in their current location.

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. On Tableau Server, open a command prompt as an administrator and navigate to the server’s bin directory. For example:

   C:\Program Files\Tableau\Tableau Server\10.5\bin

2. Type the following command:

   tabadmin importsite_verified <site ID> --importjobdir <PATH>
• `<site ID>` is the site ID of the target site.

• `<PATH>` is the directory one level up from the mappings directory you used in 3. Verify that site settings are mapped correctly. If the path has a space in it, enclose the entire path in quotation marks.

For example:

```
tabadmin importsite_verified esales --importjobdir "C:\ProgramData\Tableau\Tableau Server\data\tabsvc\temp\import_esales_20140409185810071"
```

If there is an error importing, see the `tabsvc-tabadmin-java.log` for more information. Some errors are non-fatal and can be ignored by using the `--no-verify` and `--continue-on-ignorable-errors` options. See `import-site_verified`.

3. Open the new site and confirm that everything was imported as you expected.

Mapping file content reference

The following tables list the columns in each of the mapping files created when you run the `importsite` command.

CSV file name: mappingsDomainMapperForGroups

<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>A user group name on the source site.</td>
</tr>
<tr>
<td>source_domain_name</td>
<td>No</td>
<td>The user authentication type on the source site: either local (for local authentication) or a domain name (for Active Directory).</td>
</tr>
<tr>
<td>target_domain_name</td>
<td>Yes*</td>
<td>The user authentication type on the target site: either local for local authentication, or</td>
</tr>
</tbody>
</table>
a domain name (such as example.com or example.lan) for Active Directory.

*For the **All Users** group, keep the **target_domain_name** value set to **local**, even if your target server is configured for Active Directory user authentication. 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 <strong>Refresh Extract</strong>, for extract refreshes, or <strong>Subscriptions</strong>, 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 target site. You can edit this value. For example, if the schedule is named <strong>Friday Update</strong> on the source site you can rename it <strong>Friday Refresh</strong> on the target site.</td>
</tr>
<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 user authentication type on the source site:  <strong>local</strong>, for local authentication, a domain name (such as example.com) for Active Directory, or <strong>external</strong> (for a Tableau Online site).</td>
</tr>
<tr>
<td>target_name</td>
<td>Yes</td>
<td>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</td>
</tr>
</tbody>
</table>
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 user authentication type on the target site: either **local**, for Local Authentication, or a domain name (such as example.com) for Active Directory. |

CSV file name: MappingsScheduleRecurrenceMapperWithAutoCreation

This file does not require updates.

## 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 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 tabadmin sitestate 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**.

Enable Users to Save Revision History

Revision history enables your users to keep copies of workbooks and data sources (content resources) that change. Each time someone saves an update to a content resource, Tableau Server creates a new version and stores it with older versions.

Server administrators can enable revision history on a per-site basis, for all of its workbooks and data sources. You can determine the number of revisions saved in history. Revision history is enabled by default, with a limit of 25 revisions for each content resource.

Authorized users can take actions on earlier versions of their content. By enabling revision history, you give everyone the confidence to experiment with the content, knowing that older versions are always available.
**Note:** This topic applies to Tableau Server. Workbook revision history (and not data source revision history) is enabled on all Tableau Online sites. Users can save up to 10 revisions.

### Required permissions

To access revision history, a user must have a site role of **Publisher**, plus the following permissions:

- **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 number of revisions allowed

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 online help.

**Customize a Site**

You can customize how a site on Tableau Server looks to personalize it for your company or group. You can perform these customizations:

- Change the server name that appears in the browser tab, tooltips, and messages.
- Change the logos that appear in the web environment.
- Set the language used for the web environment and the locale used for views.
- Install custom fonts on Tableau Server and client computers that connect to Tableau Server.

Administrators and project leaders can also add images for projects in thumbnail view.

**Change the Name or Logo**

You can customize the Tableau Server look and feel to brand it for your company by changing the server name and by using a custom logo.
Your custom name appears on browser tabs and in a tooltip when users hover over the home logo in the upper left corner of the main page. The customizable logo appears in the sign-in page and the server page header. (Note that some references to Tableau Server cannot be changed, such as the logo on browser tabs and the phrase "Tableau Server" in the copyright notice.)

A - The Tableau logo for the browser window tab cannot be changed.

B - The server name can be changed using `tabadmin customize name`.

C - The header logo can be changed using `tabadmin customize header_logo`.

Change the server name

The server name is displayed in tooltips, messages, and on the browser window tab. The following example shows a custom name displayed as a tooltip and on the browser window tab.
1. Open a command prompt as an administrator and type the following:

   ```
   cd "C:\Program Files\Tableau\Tableau Server\10.5\bin"
   ```

2. Type the following command:

   ```
   tabadmin customize name "new_name"
   ```

   Replace `new_name` with your custom name, as in the following example:

   ```
   tabadmin customize name "My Company"
   ```

   **Note:** To change to a name that includes Unicode characters, identify the hex encoding for each Unicode character and add \u before each hex value. For
example, for the two-character string 测试, type the command `tabadmin customize name "测试"`.  

3. Type the following command to restart the server so that the change takes effect:

   `tabadmin restart`

Change the logo

You can customize the logo that appears on the Tableau Server sign-in page, the header logo in server pages, and the small logo that appears in the upper left when a view is being edited in web authoring. The name "Tableau" is part of this logo. It cannot be changed independently of the logo.

The image files you use should be in GIF, JPEG, or PNG format.

The header logo image can be up to 160 by 160 pixels, but not smaller than 32 by 32 pixels. For best results use an image that is 125 by 35 pixels. If the image is larger than 160 by 160 pixels, it is clipped.

The sign in logo image can be a maximum of 3000 by 3000 pixels.

**Note:** The background colors differ in these locations, so your logo might look different depending on where it appears in the server interface.

Header logo

![Header logo image](image-url)
This is the default header logo.

This is what a custom header logo might look like.

Sign-in logo

This is the default sign-in logo.
This is what a custom sign-in logo might look like.

Customize a logo

1. Open a command prompt as an administrator and type the following command:

   cd "C:\Program Files\Tableau\Tableau Server\10.5\bin"

2. Type one of the following commands, depending on which logo you want to set. Substitute your own image file for C:\My Pictures\logo.png.

   tabadmin customize header_logo "C:\My Pictures\logo.png"
   tabadmin customize sign_in_logo "C:\My Pictures\logo.png"
Note: If an image for the logo or the header logo is larger than 160 by 160 pixels, it is clipped.

For more information on the `tabadmin customize` command, see customize.

3. Type the following command to restart the server so that the change takes effect:

   `tabadmin restart`

Restore the default name or logo

1. Open a command prompt as an administrator and type the following command:

   `cd "C:\Program Files\Tableau\Tableau Server\10.5\bin"

2. Type the following command:

   `tabadmin customize parameter -d`

   For `parameter`, specify `name, logo, header_logo, or sign_in_logo`.

3. Type the following command to restart the server so that the change takes effect:

   `tabadmin restart`

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 (see Server Settings (General), 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.
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.
- 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 in Windows, see the Microsoft Knowledgebase.
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, even if the schedule contains only one refresh task. Serial schedules run on only one backgrounder process.
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.

See also

Enable Extract Refresh Scheduling and Failure Notification

Automate Refresh Tasks

Enable Extract Refresh Scheduling and Failure Notification

Before your publishers can schedule extract refreshes, you must enable scheduling on the server.

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 as a server administrator, and select Settings.

2. 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.

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.

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—either refreshing extracts 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. 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.
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

**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. There is no distinction between extract refreshes, subscriptions, and data-driven alerts.

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 per-
formance-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 backgrounder 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 tabadmin set option to turn off workbook caching after a scheduled refresh:

```
backgrounder.externalquerycachewarmup.enabled
```

For more information on how to use and apply tabadmin set options, see tabadmin.

**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 tabadmin 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:

- **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 View or Edit Permission Rules and User Permissions.

- **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 Server 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, 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 data-driven alerts, users need access to related databases and views. See this list of requirements for details.


**In this article**

- Configure the server to send subscription emails
- Enable subscriptions in a site
- Test subscriptions in a site
- Manage all user subscriptions
Configure the server to send subscription emails

1. In the Tableau Server Configuration utility, click the Alerts and Subscriptions tab, and then select Enable users to receive emails for subscriptions to views.

2. Configure SMTP Setup.

Enable subscriptions in a site

As an administrator, you decide which sites allow subscriptions.

1. At the top the browser window, click Settings.

2. Select Allow users to subscribe to workbooks and views.

3. Optionally, select Allow content owners to subscribe other users to workbooks and sheets, or enter an email "From" address and message footer.

   A site's "From" address and message footer are also used in emails for data-driven alerts.

4. 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.)

Set Up a Server 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.

In this article

Configure email for data-driven alerts
Manage all data-driven alerts in a site
Disable data-driven alerts for a site
Control how often the server checks data-driven alerts
Track the server’s alert-checking process
Identify and fix failing alerts

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**.

**Tip**: To entirely disable data-driven alerts at the server level, change the `tabadmin set` option, "features.AlertOnThresholdCondition", to false.

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 `tabadmin set` option, "dataAlerts.checkIntervalInMinutes".

Regardless of the `dataAlerts.checkIntervalInMinutes` 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 use the "Check If Data Alert Condition is True" task to proactively identify failing alerts that users are unaware of. 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.)
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).

Performance
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.

• **Alerts.** 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. Confirm that **Refresh Less Often** on the Data Connections tab of the Configuration dialog box is selected.

**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 tabadmin 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 consumes a good deal of session memory. Instead, you can end sessions when users move away from a view by changing the value of the `vizqlserver.clear_session_on_unload` setting to `true` (the default is `false`).

**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.

**Tableau Server Alerts**

Alerts are email notifications that you receive when something happens on Tableau Server. You can set up alerts for when the server is running out of disk space and for when server processes stop or start. These conditions often mean that there is an immediate problem.

**Note:** We discuss alerts in this section as a tool for getting information about server health. But as an entirely separate benefit, users can also make use of alerts. After you set up alerts, your users can subscribe to views to periodically receive a snapshot of views they are interested in on a recurring basis.

To send alerts, Tableau Server must connect to a mail server, also known as a Simple Mail Transfer Protocol (SMTP) server. An SMTP server is a service that you can send outbound email messages to. It then relays the messages to whoever they’re addressed to. (It doesn’t handle incoming email.) To set up alerts, you must configure Tableau Server to communicate with your SMTP mail server.

**SMTP information you’ll need**

Many organizations already have an SMTP server in-house. Before you continue, ask your IT department if there is an SMTP server that you can use.

Here’s the SMTP server information that you need from your IT department:
• The server address. This is often something like smtp.example.com or mail.example.com, but other addresses are also possible.

• The port. This is 25 for most servers.

• A user name.

• A password.

Some servers don't require a user name or password because they are only meant for internal use.

You'll also need to decide on a from address for the alerts that the server sends. When people receive an alert email from Tableau Server, this is the name that's on the from line of the message. Because alerts are simply informational, you generally don't need to worry about who's on the from line, so people use addresses like no-reply@example.com or tableau-admin@example.com.

After you get the SMTP server information from your IT department, you can use the Tableau Server Configuration utility to set up alerts.

Step 1: Stop the server

1. Stop Tableau Server. (In the Windows Start menu, search for Stop Tableau Server.)

2. In the Windows Start menu, search for Configure Tableau Server.

Step 2: Configure SMTP information for Tableau Server

1. In the Tableau Server Configuration utility, click the SMTP Setup tab.

2. Enter the information that you received from your IT department.

3. In the Send email from box, enter the email address that you want all server emails to be sent from. For example, you might enter tableau_admin@example.com or no-reply@example.com.
4. In the **Send email to** box, enter the email address or addresses that you want server-health emails to be sent to. For example, you might enter your own email address and the email address of your IT person.

![Tableau Server Configuration](image)

5. Click **OK**.

6. Start Tableau Server (in the Windows Start menu, search for **Start Tableau Server**).

**Step 3: Set up alerts**

In the Tableau Server Configuration utility, click the **Alerts and Subscriptions** tab. We recommend that you select all the checkboxes on this tab to enable all alerts. You’ll know that alerts are working when you restart Tableau Server and receive an email.
If you do select all the check boxes, here are the alerts that get activated.

Subscriptions to views

Users can periodically receive a snapshot of views that they're interested in. This can be useful if your users want to see information about views on a recurring basis. For example, users can get a view in their inboxes every week.

Server component events

Anytime that server processes stop or that the server restarts unexpectedly, you should investigate the cause of the restart. For example, you may discover that the Windows Server computer is configured to restart automatically after Windows updates—in which case you may want to schedule updates for off-peak hours.
Low disk space

You can receive a notification when the disk space on the server computer falls below a threshold that you specify. As a general rule, we recommend that the server computer maintain at least 20% free disk space. The farther that the disk space falls below this threshold, the more likely that the server’s performance will be affected. Eventually, the server may even stop responding.

Step 4: Restart the server

On the Windows Start menu, click **All Programs > Tableau Server 10.5 > Start Tableau Server**. If alerts are configured correctly, Tableau Server sends an email titled "Multiple services on your-server are UP."

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 an external performance monitoring tool. (For the purposes of this section, we’ll use Windows Performance Monitor as an example, because it’s included with Windows Server.) To collect additional load time data and background task data, you can connect to the Tableau Server repository.

After you’ve collected the performance data that you want to analyze, you can use the sample workbook included in this section as a starting point for analyzing your performance data. To make it easier to analyze your performance data in one place, you can then publish the views that you create to Tableau Server as custom administrative views.

For more information on the built-in administrative views, see Administrative Views.

For more information on custom administrative views, see Create Custom Administrative Views.

**Note:** To use the sample workbook and to publish views to Tableau Server, you must have Tableau Desktop.

### Collect Data with Windows Performance Monitor

To monitor resource usage and server processes, you can use Windows Performance Monitor (PerfMon), which is included with Windows Server. Use PerfMon to gather detailed performance information, including how often the CPU is being used, how much memory is being used, information about each Tableau Server process, and more.

For more information about what each Tableau Server process does, see Tableau Server Processes.

**Disclaimer:** This information refers to a third-party product. This example is not an endorsement of this product over any other competing products.
Before you can use PerfMon, you set up a data collector set, which is how PerfMon stores the data that it collects. To collect information about Tableau Server processes with PerfMon, Tableau Server must be running when you create the data collector set. The data that you collect in PerfMon are often referred to as performance counters.

Step 1: Create a new data collector set

1. Click the Windows Start menu and search for "performance".
2. Right-click Performance Monitor and then click Run as administrator.
3. In the left pane, click Data Collector Sets.
4. In the right pane, right-click User Defined, click New, and then click Data Collector Set.
5. In the Create new Data Collector Set wizard, enter a name for the data collector set. For example, you might enter Tableau Server Performance.
6. Select Create manually (Advanced) and then click Next.
7. Under Create data logs, select Performance counter, and click Next.

Step 2: Select performance counters

1. Set the sample interval to 30 seconds.
2. Click Add.
3. Select performance counters from the list.

The following table lists some performance counters that we recommend for tracking Tableau Server performance.

<table>
<thead>
<tr>
<th>Category</th>
<th>Performance Counters</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Logical Disk</td>
<td>Current Disk</td>
<td>The number of outstanding write requests and the</td>
</tr>
<tr>
<td>Category</td>
<td>Performance Counters</td>
<td>Notes</td>
</tr>
<tr>
<td>------------------</td>
<td>--------------------------------------------------------------------------------------</td>
<td>----------------------------------------------------------------------</td>
</tr>
<tr>
<td></td>
<td>Queue Length</td>
<td>amount of bytes read and written to the server’s hard disk. Select these counters for the disk on which you installed Tableau Server (referred to as the <em>instance</em> in PerfMon).</td>
</tr>
<tr>
<td></td>
<td>Disk Read Bytes/sec</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Disk Write Bytes/sec</td>
<td></td>
</tr>
<tr>
<td>Memory</td>
<td>% Committed Bytes in Use</td>
<td>The percentage of virtual memory in use, and the amount of memory available in megabytes.</td>
</tr>
<tr>
<td></td>
<td>Available MBytes</td>
<td></td>
</tr>
<tr>
<td>Processor</td>
<td>% Processor Time</td>
<td>The percentage of time that the processor spends active, and the percent of processing capacity being used by the processor.</td>
</tr>
<tr>
<td>Information</td>
<td>% Processor Utility</td>
<td></td>
</tr>
<tr>
<td>Process</td>
<td>% Processor Time</td>
<td>The percentage of processing capacity being used by a particular process, and the amount of memory reserved for the process. Select these counters for the following processes (referred to as <em>instances</em> in PerfMon):</td>
</tr>
<tr>
<td></td>
<td>Private Bytes</td>
<td></td>
</tr>
<tr>
<td></td>
<td>backgrounder <em>(Backgrounder)</em></td>
<td></td>
</tr>
<tr>
<td></td>
<td>dataserver <em>(Data server)</em></td>
<td></td>
</tr>
<tr>
<td></td>
<td>redis-server <em>(Cache server)</em></td>
<td></td>
</tr>
<tr>
<td></td>
<td>hyperd <em>(Data engine)</em></td>
<td></td>
</tr>
<tr>
<td></td>
<td>vizqlserver <em>(VizQL Server)</em></td>
<td></td>
</tr>
</tbody>
</table>

To select each performance counter:
1. Double-click to select a category in the drop-down list.

2. Select the performance counter or counters that you want to use.

3. Under **Instances of selected object**, if appropriate, select the process (or instance) that you want to collect information about.

4. Click **Add**.

5. Click **OK** and then click **Next**.
Step 3: Save the data collector set

1. Browse to the directory where you want to store the data, and then click **Next**.

   **Important:** You must store the data in a place that's accessible by Tableau. For example, you might want to store the data on a network drive. If you don't have a network drive mapped, right-click **This PC** and select **Add a Network Location**.

2. Click **Finish**.

3. In the left pane of the main **Performance Monitor** window, select the data collector set that you created.

4. In the right pane, right-click the performance counter **DataCollector01** and then click **Properties**.

5. Select **Comma separated** as the log format and then click **OK**.

Step 4: Run the data collector set

In the left pane, right-click the name of the data collector set that you created and click **Start**. The Windows Performance Monitor tool starts monitoring your server and storing information in the location that you specified.

**Collect Data with the Tableau Server Repository**

Before you start analyzing the resource usage data that Windows Performance Monitor (PerfMon) collects, connect to the Tableau Server repository to get additional data about load times and background tasks. The Tableau Server repository is a PostgreSQL database that stores data about all user interactions, extract refreshes, and more.

Looking for Tableau Server on Linux? 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. Alternatively, if you are only interested in performance data, you can use the preselected database tables in the sample performance workbook. For more information about the sample performance workbook, see Analyze Data with the Sample Performance Workbook.

Enable access to the Tableau Server repository

To access the Tableau Server repository, you need to enable access to the database by using tabadmin commands.

1. Open a command prompt as an administrator and type the following:
   
   ```
   cd "C:\Program Files\Tableau\Tableau Server\10.5\bin"
   ```

2. Enter the following command to enable external access to the database for the tableau user or the readonly user:
   
   ```
   tabadmin dbpass --username [tableau|readonly] password
   ```

   For example, to enable access for the readonly user with a password of p@ssword, use this command:
   
   ```
   tabadmin dbpass --username readonly p@ssword
   ```

   **Note:** If no user is specified, the dbpass command enables access for the tableau user.

3. Restart Tableau Server by using this command:
   
   ```
   tabadmin restart
   ```
**Note:** If you later decide that you want to disable remote access to the Tableau Server repository, use the `tabadmin dbpass --disable` command. The `--disable` option disables external access to the repository. This will not disable access from localhost. For more information, see [tabadmin Commands](#).

**Connect to the Tableau Server repository**

This section describes how to connect to a custom set of tables from Tableau Server repository. If you want to use the sample performance workbook, see [Analyze Data with the Sample Performance Workbook](#) and follow the steps to edit the connection information.

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.

   **Note:** The `tabadmin dbpass` command does not open any ports in the firewall. You may need to manually open the port in any firewall between your external client and the Tableau Server database.

3. Specify **workgroup** as the database to connect to.

4. Connect using the user and the password you specified.

5. Click **Connect**.
6. 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 `_datasources`. 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.

7. Click **Go to Worksheet**.
Analyze Data with the Sample Performance Workbook

**Important:** Before reading this topic, you need to complete the steps in Collect Data with Windows Performance Monitor and Collect Data with the Tableau Server Repository.

To get started analyzing the data that you collect with Windows Performance Monitor (PerfMon) and with the Tableau Server repository, you can download and use the sample performance workbook that Tableau provides. The sample workbook contains worksheets for some of the most important performance indicators, including CPU and memory utilization by process, disk activity, view load times, and more. After you download the workbook, use it as a starting point for your data exploration and extend it to meet your needs.

Download the sample performance workbook

1. In the workbook that's embedded below, click the Download button in the lower-right corner of the workbook.

2. Navigate to the directory where you downloaded the workbook and double-click on the workbook to open it in Tableau Desktop.

Edit the connections to the Tableau Server repository

After you open the workbook, you must edit the data connections to point to the PostgreSQL repository for your installation of Tableau Server.

The sample workbook includes the following data source connections:

- **Background Tasks**—a connection to the Tableau Server repository that joins the `background_tasks` table and the `sites` table.

- **Historical Events**—a connection to the Tableau Server repository that joins the `historical_events`, `hist_users`, and `historical_event_types` tables.

- **Resource Usage**—a connection to the data you collect from PerfMon.
• **Sessions and Load Times**—a connection to the Tableau Server repository that joins the _users, _http_requests, _sessions, and _sites tables.

1. From any of the sheets in the workbook, right-click the **Background Tasks** data source in the **Data** pane, and then click **Edit Data Source**. (Note that you might have to right-click on a dashboard and click **Unhide All Sheets**.)

   ![PostgreSQL Connection](image)

   2. Replace the server address with the hostname or IP address of your server.

   3. Enter the password that you set for the **readonly** user.

   4. Click **OK**.

   5. Repeat the steps above for the **Sessions and Load Times** data source.
Edit the connection to the PerfMon data

1. Click the Data Source tab.

2. In the data source list, select Resource Usage.

![Resource Usage](image)

3. In the menu, click Data > Resource Usage > Edit Connection.

4. Navigate to the directory where you stored the data from PerfMon.

5. Select the file and click Open.

Replace references to PerfMon fields

Because the data that you collect from PerfMon includes references to specific host names, you must also replace the references in the workbook for specific field names. For example, when you click on the CPU worksheet, you might notice that most of the fields have a red exclamation mark next to them. This indicates that the field is missing from the new data source.

![Example Field](image)

To map the fields in your data to the fields in the workbook, follow these steps:

1. Navigate to the CPU worksheet or to any of the worksheets that use the Resource Usage data source.

2. Right-click a field in the Measures list that has a red exclamation mark next to it.

3. Click Replace references.
4. Select the corresponding field in the dialog.

For example, you might replace references to `\YOUR-SERVER\LogicalDisk (C:)\Current Disk Queue Length`.

Update calculations

The sample workbook includes calculations that aggregate processor utilization for multiple processes. To view data for all of the processes on your server, you must edit the calculations to include additional instances of server processes.

For example, the **VizQL Server CPU %** calculation includes a reference to one process:

```
[[\YOUR-SERVER\Process(vizqlserver)\% Processor Time]
```

However, if your server runs more than one VizQL Server process, then you must aggregate the additional process in the calculation. For example, you might enter the following:

```
[[\YOUR-SERVER\Process(vizqlserver)\% Processor Time] +
[[\YOUR-SERVER\Process(vizqlserver#1)\% Processor Time]
```

Remove the date filter from the extract

To limit the amount of data bundled in the sample performance workbook (and reduce the size of the download), the workbook includes a date filter on the included extract. The visualizations in the workbook will display as blank until the date filter is removed.

1. Click the **Data Source** tab.

2. In the upper-right, click **Edit**.

![Connection Live Extract Edit](extract_icon.png)

Extract includes subset of data

3. Select the data source filter.
4. Click **Remove**.

Publish to Tableau Server

Optionally, when you finish updating your performance workbook, you can publish it to Tableau Server so that the views in the workbook are accessible from the Tableau Server interface.

**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 Windows Performance Monitor? 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.

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.

**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.

**Note:** This topic uses the sample performance workbook from the monitoring section. For more information, see Analyze Data with the Sample Performance Workbook.

- 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.

1. Stop Tableau Server and open the Tableau Server Configuration utility.

2. Click the **Servers** tab.

3. Click **Edit**.
4. Increase the number of VizQL server processes by one.

5. Restart Tableau Server.

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 tabadmin to reduce the `vizqlserver.session.expiry.timeout` setting.

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.

1. Stop Tableau Server and open the Tableau Server Configuration utility.

2. Click the Data Connections tab.
3. Select **Refresh less often** as the caching option.

4. Click **OK**.

5. Restart Tableau Server.

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 About Client-Side Rendering.

About Client-Side Rendering

When you navigate to a view in Tableau Server, the processing required to display the view (the rendering) can either be performed by your client web browser or by Tableau Server depending on the complexity of the view. The complexity of the view is determined by the number of marks, rows, columns, and more. If a view is less complex, then it is faster for your web browser to render the view than it is to send a request to Tableau Server. If a view is more complex, then it is faster to send a request to Tableau Server and take advantage of the server's computing power.

As a server administrator, you can configure when client-side rendering happens both for web browsers on your computer and web browsers on mobile devices by adjusting the complexity threshold.
Requirements

- **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 is used by client-side rendering.

- **Polygons and the page history feature**: If a view uses the polygon mark type or the page history feature, server-side rendering is performed, even if client-side rendering is otherwise enabled.

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 in your computer’s web browser than in your mobile device’s web browser. You can adjust how much client-side rendering happens for computers and mobile devices with the complexity thresholds. You might want to adjust the complexity thresholds if you notice that views display slowly on mobile devices. Alternatively, you might want to increase the thresholds 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 computer web browsers, use the following command:

```
tabadmin set vizqlserver.browser.render_threshold [new value]
```

By default, the complexity threshold for web browsers on mobile devices is 60. To adjust the complexity threshold for web browsers on mobile devices, use the following command:

```
tabadmin set vizqlserver.browser.render_threshold_mobile [new value]
```

For example, to change the mobile threshold to 40, you might enter the following command:

```
tabadmin set vizqlserver.browser.render_threshold_mobile 40
```

For more information on how to use `tabadmin`, see `tabadmin`. 
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:

`tabadmin set vizqlserver.browser.render false`

For more information on how to use `tabadmin`, see `tabadmin`.

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.
Note: This topic uses the sample performance workbook from the monitoring section. For more information, see Analyze Data with the Sample Performance Workbook.

- When to optimize for extracts
- Ways to optimize for extracts

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.

To increase the number of backgrounder processes, complete the following steps.

1. Stop Tableau Server and open the Tableau Server Configuration utility.
2. Click the **Servers** tab.
3. Click **Edit**.
4. Increase the number of Backgrounder processes by one.
5. Restart Tableau Server.

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.

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 VizQL 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:** 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. Extract heavy sites benefit from isolating the Backgrounder process on its own machine.

• **Frequent extract refreshes:** Refreshing an extract is a CPU-intensive task. 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.

• **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
- 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 Linux? 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:
3. Under Workbook Performance Metrics, select **Record workbook performance metrics**.

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:

   http://10.32.139.22/#/views/Coffee_Sales2013/USSalesMarginsByAreaCode?:iid=1

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?:record_performance=yes&: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**
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:
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 Resources

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 generation tool that you can use to understand how Tableau Server responds to user interactions over time. Use TabJolt to establish a baseline for server performance and test deployments before pushing them to production 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.

- **TabMon.** A monitoring tool that uses Windows Performance Monitor and Java Management Extensions to record performance data about Tableau Server to a PostGreSQL database.

- **Microsoft System Center.** A set of server management products for monitoring, configuration, automation, and more.
- HP Sitescope. An agentless application monitoring tool.


- Splunk. A tool for monitoring and analyzing machine data, including logs.

- Graylog. An open-source log management tool.

About Client-Side Rendering

When you navigate to a view in Tableau Server, the processing required to display the view (the rendering) can either be performed by your client web browser or by Tableau Server depending on the complexity of the view. The complexity of the view is determined by the number of marks, rows, columns, and more. If a view is less complex, then it is faster for your web browser to render the view than it is to send a request to Tableau Server. If a view is more complex, then it is faster to send a request to Tableau Server and take advantage of the server’s computing power.

As a server administrator, you can configure when client-side rendering happens both for web browsers on your computer and web browsers on mobile devices by adjusting the complexity threshold.

Requirements

- **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 is used by client-side rendering.

- **Polygons and the page history feature**: If a view uses the polygon mark type or the page history feature, server-side rendering is performed, even if client-side rendering is otherwise enabled.

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 in your computer’s web browser than in your mobile
device's web browser. You can adjust how much client-side rendering happens for computers and mobile devices with the complexity thresholds. You might want to adjust the complexity thresholds if you notice that views display slowly on mobile devices. Alternatively, you might want to increase the thresholds 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 computer web browsers, use the following command:

tabadmin set vizqlserver.browser.render_threshold [new value]

By default, the complexity threshold for web browsers on mobile devices is 60. To adjust the complexity threshold for web browsers on mobile devices, use the following command:

tabadmin set vizqlserver.browser.render_threshold_mobile [new value]

For example, to change the mobile threshold to 40, you might enter the following command:

tabadmin set vizqlserver.browser.render_threshold_mobile 40

For more information on how to use tabadmin, see tabadmin.

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:

`tabadmin set vizqlserver.browser.render false`

For more information on how to use tabadmin, see tabadmin.
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

Quick Start: Disk Space Alerts

You can configure Tableau Server to monitor free disk space on computers running Tableau Server, and to send alerts when free space drops below thresholds that you define. If you choose to have Tableau Server save historical usage information, this is available to Tableau Server administrators through one of the Administrative Views.

1 Configure SMTP

Before you can configure alerts for disk space usage, you need to configure Tableau Server for SMTP on the SMTP Setup tab in the Tableau Server Configuration utility.
For more information about how to configure SMTP in Tableau Server, see Configure SMTP Setup.

2 (Optional) Change the historical disk usage data option

By default Tableau Server is configured to save data about disk space usage. If you do not want to save this data, clear the **Record disk space usage information, including threshold violations** box.
Note: You do not need to save disk space usage information to receive alerts about low disk space, but if Tableau Server is not saving disk space usage data, you cannot view historical disk space usage in Administrative Views.

3 Configure alerts

Tableau Server can send alerts to let you know when space on one of the Tableau Server nodes drops below the warning and critical thresholds of the entire disk. Tableau Server continues to send alerts at the frequency specified in **Send email alert every** as long as disk space remains below the warning threshold.

To receive email alerts when free disk space falls below either of the two thresholds, select **Send alerts when unused drive space drops below thresholds**:
4 Change alert thresholds and frequency

By default, the warning threshold is set to 20% and the critical threshold is set to 10%. As long as the free disk space remains below a threshold, Tableau Server will continue to send alerts at the frequency you specify in Send email alert every. You can change these values.

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.
1. In the Tableau Server Configuration utility, click the **SMTP Setup** tab.

2. Under **SMTP Server**:
   
   a. Enter the name of your SMTP server.
   
   b. (Optional) If your account requires it, enter a user name and password for your SMTP server account.
   
   c. If you are not using the default SMTP port 25, change the SMTP port value.
   
   d. Uncheck **Enable TLS** so the connection to your mail server is unencrypted. (Encrypted SMTP connections are not supported for alerts or subscriptions.)

![Tableau Server Configuration](image)

3. For **Send email from**, enter the email address that will send an alert if there’s a system failure. The email address must have valid syntax (for example, ITalerts@bigco.com 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 **Send email from** address on a per-site basis. For more information, see What is a site.

4. For **Send email to**, enter at least one email address that will receive the alerts. If you enter multiple addresses, separate them with commas.

5. For **Tableau Server URL**, enter `http://` or `https://`, followed by the name or IP address of the Tableau server. This value will be used for the footer of subscription emails.

6. Click **OK**.

After configuring SMTP and completing the steps to configure alerts (Configure Server Alerts), a **restart of server** will trigger an email alert. This confirms that you have set up alerts correctly.

### Configure Server Alerts

On the **Alerts and Subscriptions** tab of the Tableau Server Configuration utility, you can configure the following server alerts:

- Email alerts for system failures
- Disk space usage:
  - Recording usage history
  - Email alerts when space crosses or remains below pre-configured thresholds

In the same part of the utility, you can also enable emails for subscriptions to views. For detailed information, see Set Up a Server for Subscriptions.
Alerts for system failures

Tableau Server can send email alerts to server administrators when there is a system failure.

When you configure alerts, Tableau Server sends an email to the recipients listed in Send email to on the SMTP Setup tab any time that the data engine, repository, or gateway server processes stop or restart, or any time the primary Tableau Server stops or restarts. If you are running a single-server installation (all processes on the same machine), 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 (see Configure for Failover and Multiple Gateways), 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.

To configure email alerts for system failures

1. On the Alerts and Subscriptions tab of the Tableau Server Configuration utility, select Send email alerts for server component up, down, and failover events.

2. Click OK.
Disk space monitoring

If Tableau Server is configured to monitor free disk space and send alerts about low disk space, when space on any node in a server installation drops below the configured thresholds, Tableau Server sends an email to the recipients listed in **Send email to** on the **SMTP Setup** tab.

Disk space usage

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.

To configure Tableau Server to record disk space usage

1. On the **Alerts and Subscriptions** tab of the Tableau Server Configuration utility, select **Record disk space usage information, including threshold violations**.

2. Click **OK**.

You can configure Tableau Server to send email alerts when disk space usage on any node crosses a threshold, or remains below the threshold.
To configure email alerts for low disk space

1. On the Alerts and Subscriptions tab of the Tableau Server Configuration utility, select Send alerts when unused drive space drops below thresholds.

2. In Warning threshold, enter the percentage of free disk space that Tableau Server should use as a warning threshold.

   If free disk space on any node in your Tableau Server cluster drops below this percentage, Tableau Server sends a warning alert email. Alerts continue until free disk space rises above the threshold. To configure the frequency of alerts, see Step 4 below.

3. In Critical threshold, enter the percentage of free disk space that Tableau Server should use as a critical threshold.

   If free disk space on any node in your Tableau Server cluster drops below this percentage, Tableau Server sends a critical alert email. Alerts continue until free disk space rises above the threshold. To configure the frequency of alerts, see Step 4 below.

4. In Send email alert every, enter the number of minutes for how often Tableau Server should send an alert.

5. Click OK.
Troubleshoot Disk Space Usage on Tableau Server Nodes

When available disk space on a Tableau Server primary or worker node is low, performance can be degraded. If free space falls too low, Tableau Server may begin to perform erratically. To monitor free disk space, configure Tableau Server to save disk usage information (this is on by default) and, if desired, enable alerts about low disk space. For more information, see Quick Start: Disk Space Alerts.

**Note:** Disk space monitoring measures free disk space on each server node. Available space may be impacted by programs or processes that are not a part of Tableau Server.

If you find that your Tableau Server installation is running into free disk space limitations, you should take steps to make more space available. This topic suggests some ways you can do that.

Viewing Disk Usage on Tableau Server Nodes

When disk space usage monitoring is enabled (this is the default), server administrators can use the Server Disk Space administrative view to see current disk space usage, and one month of usage data on your Tableau Server nodes. Use this view to help you determine whether one of your server nodes is experiencing a jump in space usage, or if space usage has increased over time.

Cleaning Up Tableau Server-Related Files

To minimize server space used by Tableau Server, you can clean up unnecessary files.

Use the `tabadmin cleanup` command to remove log files, temporary files, and unneeded entries in the PostgreSQL database. If you want to save the logs before you clean them up, you can make an archive. For more information, see Remove Unneeded Files.
Once you have cleaned up log files and temporary files, you may want to use the administrative views to determine which workbooks and data sources are taking up the most space on your server, and whether any of these is not being used. For more information, see Administrative Views.

Identifying and Cleaning Up Other Files

There are a number of tools, like WinDirStat, you can use for viewing disk usage and doing cleanup.

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.

- Shows server and site activity for Tableau Server.
- Shows site activity for Tableau Online.

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 **Status**.

![Site Status](image)

**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. For information on how to optimize the server, see Optimize for User Traffic.

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:

- Display less information in each view.
- Break up views.
- Reduce the number of filters.
- Use data extracts.
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 time line 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 workbooks and data sources.
- **Subscriptions**—This shows counts of subscription email sent for workbooks and views.

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.

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.
- **Publish & Download**—This shows publishing and downloading of workbooks and data sources.
- **Subscriptions**—This shows counts of subscription email sent for workbooks and views.

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.

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.

Tasks can have a status of successful 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>
<tr>
<td>✱</td>
<td>Success—Server completed the task.</td>
</tr>
</tbody>
</table>

For details on a task, hover over its icon:

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 successful or error.

<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>

For details on a task, hover over its icon.
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 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. For more information on monitoring processes, see Collect Data with Windows Performance Monitor. For more information on tuning the performance of server processes, see Performance Tuning.

**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:

**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, pro-
ject, and by the size of the 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 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:

![Graph showing workbooks and data sources using the most space]

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.

For more information, see Troubleshoot Disk Space Usage on Tableau Server Nodes

**Tip:** You can have Tableau Server alert you when free disk space falls below a threshold that you specify. For more information, see Quick Start: Disk Space Alerts.
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](include.png)
![Exclude values](exclude.png)

**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:

- Keep only
- Exclude

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?
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.

**Note:** The readonly user is available in Tableau Server 8.2.5 and later.

For examples of custom administrative views, see the Tableau Community.

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 tabadmin 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 tabadmin set option wgserver.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 every time tabadmin cleanup or tabadmin backup is used. 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.

Maintenance

Database Maintenance

A Tableau Server administrator should perform regular database maintenance, monitor disk usage on the server, and clean up unnecessary files to free up space on the server. Taking these steps can help ensure that Tableau Server runs with maximum efficiency.

You can use the tabadmin command line tool to back up and restore your Tableau data, and to clean up (remove) unnecessary log and temporary files. Tableau data includes Tableau Server's own PostgreSQL database, which stores workbook and user metadata, data extract files, 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. Use the commands described in the topics below, along with the built-in Windows task scheduler to automate backing up data and cleaning up unnecessary files.
Note: You can only use backups made with the `tabadmin backup` command when restoring Tableau Server data. Database backups made in other ways, and virtual machine snapshots are not valid sources for restoring Tableau Server.

Back Up Tableau Server Data

Backing up Tableau Server using the `tabadmin backup` command is an important part of proper administration and maintenance of your server. Only backups created with the `tabadmin backup` command can be used if you need to restore Tableau Server data.


The frequency of your backups depends on your environment, including how much use the 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 use the backup to restore Tableau Server. Keep this in mind when you determine how often you should be backing up your system.

In addition to your regular backups, you should always create a current backup of Tableau Server before upgrading to a new version.

Starting with Tableau Server version 9.3, an option to verify the integrity of the backup was included. Use this option to make sure there is no issue with the database that would result in your backup not being usable. For more information about the verify option, see Verify the Tableau Postgres Database.

Note: You should run `tabadmin backup` and other commands while logged in as the Run As user account for Tableau Server, or using an account that has administrative rights that include Modify permissions. For more information, see Run As Account Settings to Confirm in the Tableau Server Help.
In this article

- Creating a regular backup
- Creating a pre-upgrade backup
- Scripting the backup process

Creating a regular backup

When you back up your Tableau data regularly, you can quickly restore published workbooks, data sources, and other information if there is a system failure. How often you create a backup depends on how heavily your Tableau Server is used. The more activity there is, the more often you need to back the server up.

Tableau Server data consists of Tableau's own PostgreSQL database, which contains workbook and user metadata, data extract (.tde or .hyper) files, and configuration data. When you use tabadmin to create a backup, all these things are saved in a single file with a .tsbak extension. If you are running a distributed installation of Tableau Server you create the backup on the primary, and data from all the nodes is backed up.

For safety, after you create the backup, store the .tsbak file on a computer that is not a part of your Tableau Server installation.

**Note:** Running the `backup` command also removes Tableau Server log files older than seven days as well as some of the information displayed in certain Tableau Server Administrative Views.

1. Open a command prompt as an administrator and navigate to the bin directory. For example:

   ```
   cd "C:\Program Files\Tableau\Tableau Server\10.5\bin"
   ```

2. Create a backup file by typing `tabadmin backup <filename>`, where `<filename>` is the name or location and name of your backup file. Beginning with version
9.3, include the \texttt{--v} option to verify the integrity of the backup.

Starting with version 8.1, there is no need to stop the server before you create the backup.

For example:

\texttt{tabadmin backup tabserver --v}

or

\texttt{tabadmin backup C:\backups\tableau\tabserver --v}

You can also optionally use \texttt{--d} to append the current date to the file name.

Add \texttt{--t} followed by a path, to specify a location for temporary files that are created during the backup process. The path for the temporary files is not the location where the backup file will be written. For example:

\texttt{tabadmin backup tabserver --t C:mytemp\tableau}

In the above example, the backup file \texttt{tabserver.tsbak} will be created in the Tableau Server bin directory (\texttt{C:\Program Files\Tableau\Tableau Server\10.5\bin}) not in \texttt{C:mytemp\tableau}.

\textbf{Note:} The \texttt{--v} option is available beginning with version 9.3 and verifies the integrity of the backup. After creating the backup, tabadmin verifies that the file can be used to restore the database. For more information, see Verify the Tableau Postgres Database

Creating a pre-upgrade backup

You should always create a backup before upgrading Tableau Server. Starting with version 10.0, Tableau Server Setup offers to create a backup before upgrading to a new version. If you have created a backup yourself, before the upgrade, you can choose to skip this and save time during the upgrade. You can create a backup while Tableau Server is running and
minimize the amount of time the server is unavailable during upgrade. For more on the backup option during upgrade, see Tableau Server Upgrade Backup Options.

The process for creating a pre-upgrade backup is the same as for creating regular backups, with one additional consideration for distributed installations.

The Tableau backup file (.tsbak) includes configuration information as well as data. Therefore, a backup of a distributed installation of Tableau Server will include configuration information about the worker nodes, including their IP addresses. If you don’t want this information as part of your backup (for example, because you are creating the backup for a test deployment, or will be migrating worker nodes to new hardware as part of your upgrade), you can do one of two things:

- Plan on using the --no-config option when you restore the backup file to your new installation. With this option, no configuration information is restored, including configuration information for the primary Tableau Server node.

- Remove the workers from the Tableau Server configuration before creating the backup.

**Note:** You should uninstall Tableau Server from any workers that you are not including in your new installation to avoid conflicts between the older workers and the new installation.

Scripting 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:

- Saving log files that are older than seven days if you do want to keep them, or conversely, removing log files that you don’t want to back up.

- Running the backup itself.
• Copying the backup file to a separate computer for safekeeping.

• Verifying that the server is running after the script is finished.

This section discusses `tabadmin` commands and Windows commands that you can use together to perform a backup and related tasks.

Save logs before cleanup

Performing a backup removes log files that are older than seven days. If you want to preserve the logs before the backup, run the following command before you start the backup:

```
tabadmin ziplogs -l -n -f
```

Remove log files and clear temporary folders

You can clean old log file and temporary files from your Tableau Server computer 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:

```
tabadmin cleanup
```

If you run this command while the server is running, the command performs cleanup on log files and temporary folders that are not currently in use. To remove all Tableau Server log files and clear temp folders, stop the server first. This lets the cleanup command remove log files that might currently be open. The sequence of commands for a full cleanup is this:

```
tabadmin stop
tabadmin cleanup
tabadmin start
```

Run the backup

To create the backup, use the `tabadmin backup` command:

```
tabadmin backup <backupfilename> -d -v
```

Note the following about the command:
• Include the \(-v\) command option to verify the integrity of the backup.

• Add \(-d\) to the command to include the date in the file name.

• For the <backupfilename> value, you can specify a path and filename if appropriate. If you specify only a filename, the backup file is saved in the current directory (the Tableau Server bin directory).

• If you don't specify a filename extension, Tableau adds the .tsbak extension.

Copy the backup file to another computer

As a best practice, after the backup is finished, copy the backup file and any saved log files to another location that is separate from Tableau Server. To do this, use the Windows copy command:

```
copy <original_backup_path_and_filename> <network_drive_or_other_location_path_and_name>
```

Verify Tableau Server restart

You can use script commands to verify that the Tableau Server is running after the backup process has completed and you've run a tableau start command. If the server does not start, you can have the script run a command that sends an email to the administrator. However, before checking the server status, allow enough time for Tableau Server to finish its initialization. You can do this in a script by including a timeout command in the script. In the following example, the timeout command waits 90 seconds before continuing:

```
timeout /t 90
if tabadmin status != 'RUNNING' then <code_to_email_an_alert>
```

Complete example script

The following is an example script combining logs, backup, cleanup and alerting capabilities.

```
tabadmin ziplogs -l -n -f
copy logs.zip <path_and_filename>
tabadmin backup <backupfilename> -d -v
```
copy <original_backup_path_and_name> <other_location_path_and_name>
tabadmin cleanup
timeout /t 90
if tabadmin status != 'RUNNING' then <code_to_email_an_alert>

Scheduling a scripted backup

To run backups regularly, you can use the Windows Task Scheduler to schedule when to run it.

1. Sign into Windows using the Run As user account or another account that has administrator permissions for the server computer.

2. Click the Windows Start menu and search for "schedule".

3. Right-click Task Scheduler and then click Run as administrator

4. Follow the Windows Task Scheduler wizard to complete the setup. As the command to run, specify the name of the command file that contains the backup script.

For more information, see Task Scheduler How To in the Microsoft TechNet library.

Verify the Tableau Postgres Database

Under rare circumstances, the PostgreSQL database that Tableau Server users for its repository can become corrupted. (If corruption occurs, it's often a result of a hardware problem on the computer hosting the repository.) The corruption may not be immediately obvious and may not cause the database to stop functioning, but it can impact your ability to restore a backup of the data.

To help you avoid problems due to database corruption, as a best practice you should regularly perform these tasks:

- Verify the integrity of the PostgreSQL database.
- Back up your Tableau data.
Note: The verify option is available beginning with version 9.3 of Tableau Server.

Verifying the database

You can verify database integrity while you perform a backup, or you can verify the database, or a backup of the database, as a separate step.

To verify the database during a backup, add the -v option to the backup command:

`tabadmin backup tabserver -v`

To verify the current database or a backup of the database, use the `verify_database` command:

`tabadmin verify_database`

Note: You do not need to stop Tableau Server to verify the database.

This command verifies that a backup of the PostgreSQL database can be restored successfully. If you cannot restore the database, your backups aren't useful, and Tableau Server upgrades can fail.

Verify the integrity of the Tableau PostgreSQL database using the procedure below.

1. Open a command prompt as an administrator and type the following:

   `cd "C:\Program Files\Tableau\Tableau Server\10.5\bin"`

   Note: If you are running a distributed installation of Tableau Server, perform this step on the primary computer.

2. Verify the current database or a backup of the database. You do not need to stop Tableau Server to verify the database.
• Verify the integrity of the Postgres database by typing the following:

```
tabadmin verify_database
```

• Verify the integrity of a backup file by typing the following:

```
tabadmin verify_database -f <filename>, where <filename> is the name of your backup file.
```

For example:

```
tabadmin verify_database -f c:\-
backups\tableau\tabserver\tserver.tsbak
```

• Verify the integrity of the database while creating a backup of the Tableau Server data by adding `-v` to the backup command:

```
tabadmin backup tabserver -v
```

You can optionally specify a location for temporary files that are created during the verification process.

```
tabadmin verify_database -t C:\mytemp\tableau
```

If you don't specify a location, the default Tableau temp folder is used.

**Restore from a Backup**

Use the `tabadmin 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.

**Before you begin**

Only backups created using `tabadmin backup`, or created by the Tableau Server uninstall process can be used to restore Tableau Server data. Backups created with versions earlier than 9.x cannot be restored to version 10.x or later.
When you use `tabadmin` to restore your Tableau data, the contents of the PostgreSQL database, data extracts, and configuration files 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 primary node.

We recommend verifying the following before attempting a restore:

- If you have made permission changes to the install directories, then verify that the Run As User account has access to the Tableau directories. See Verify Folder Permissions.

- Antivirus software—Antivirus software that performs full drive scans can interfere with Tableau Server. In some cases, interference from antivirus software will prohibit Tableau Server from restoring properly. If you are running antivirus software on the computer running Tableau Server, follow the recommendations in the Knowledge Base.

- Beginning with version 9.3, a `verify_database` command allows you to verify that your backup file does not have a hidden problem that will cause the restore to fail. If you have version 9.3 or later, verify the integrity of the backup using the `tabadmin verify_database` command before you restore the database. For more information, see Verify the Tableau Postgres Database.

Restore Tableau Server from a backup file

1. On the Tableau Server computer, open a command prompt as administrator.

**Opening the command prompt as an administrator**

1. Click Start or press the Windows key.

2. Type `cmd`. Results will be listed, including `cmd.exe`, the command prompt.

3. Right-click `cmd.exe` and select **Run as administrator**:

2. Go to the Tableau Server \bin folder, located here:
C:\Program Files\Tableau\Tableau Server\<version>\bin

Where <version> is your version of Tableau Server.

For example, go to the Tableau Server 10.3 bin directory by typing the following:

cd C:\Program Files\Tableau\Tableau Server\10.3\bin

3. Stop the server:

tabadmin stop

4. Restore from a backup file:

tabadmin restore <filename>

In the above line, replace <filename> with the name of the backup file you want to restore from.

**Note:** When restoring from a backup, Tableau Server prompts for the password of the Run As user account. You can type the password when prompted, or use a .txt file that contains only the password (if you are scripting the restore, for example).

To restore only the data but no configuration settings (for example, if you are moving Tableau Server to a new computer), include the **--no-config** option:

tabadmin restore --no-config <filename>

5. Restart the server:

tabadmin start

6. If you ran the tabadmin assetkeys command at any time before you created the backup file that you’re now restoring, run the following command:

tabadmin assetkeys --validate
You'll be prompted to enter the passphrase needed to re-create the custom encryption keys in use in the backup file.

**Note:** When a restore completes successfully these folders are deleted automatically, but you may find extraneous folders if you have had restore failures. If Tableau Server is running correctly with all expected data, it’s safe to remove any `tabsvc.bak-*` folders from `ProgramData\Tableau\Tableau Server\data` to free additional disk space. In Tableau Server clusters, `tabsvc.bak-*` folders are created on each machine running Tableau Server. Only remove the `tabsvc.bak-*` folders.

Do not remove the `tabsvc` folder, which is also located under `ProgramData\Tableau\Tableau Server\data`. It contains necessary Tableau Server data.

Providing the Run As user password in a file

**Note:** If you choose to store private information like passwords in a file, keep security considerations in mind. As a best practice we recommend you include a step in your process to remove the file after it is used to prevent unauthorized access.

When you restore from a backup, Tableau Server prompts you for the password of the Run As user account. If you are scripting the restore, you can provide the password in a `.txt` file that contains only the password.

For example, to restore to a new computer, use these commands:

```
tabadmin restore --no-config <backupfilename.tsbak> --password-file <passwordfile.txt>
```

```
tabadmin start
```
By default, Tableau Server looks in the bin folder for the password file. If you save the file in a different location, include the path for the file. For example:

```
tabadmin restore --no-config <backupfilename.tsbak> --password-file <c:\<location>\passwordfile.txt>
```

tabadmin start

**Recover Extracts from a Backup**

The file `uninstall-<version>.tsbak` (for example, `uninstall-10.4.tsbak`) is created as part of the uninstall process. After you upgrade to version 10.5, you can use this file to restore data extracts—for example, if you mistakenly deleted the dataengine folder during the upgrade. To use `uninstall-<version>.tsbak` to restore data extracts:

1. **Stop the server.**
2. From within your version 10.5 Tableau Server bin directory, type the following:
   
   **Windows Server 2012, Windows Server 2008, Windows 7, Windows 8:** `tabadmin restore \ProgramData\Tableau\Tableau Server\uninstall-10.4.tsbak`

   **32-bit Tableau Server installed on 64-bit Windows Server:** `tabadmin restore \Program Files (x86)\Tableau\Tableau Server\uninstall-10.4.tsbak`

   **32-bit Tableau Server installed on 32-bit Windows Server:** `tabadmin restore \Program Files\Tableau\Tableau Server\uninstall-10.4.tsbak`

**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 `cleanup` command to remove Tableau Server log files, temporary files, and rows older than seven days from the `http_requests` table of the Tableau Server PostgreSQL database. If you might need older logs for troubleshooting,
you should create a log file archive before doing the cleanup. For more information, see Archive Logs on Command Line (tabadmin).

The http_requests table contains records of every http request that comes in through the gateway, so it can grow significantly if you do not regularly clean it as part of your database maintenance. If the table gets too large it can impact performance and cause disk space issues.

To perform a cleanup, use this command:

```
tabadmin cleanup
```

You can add the `restart`, option, which is the equivalent of running `tabadmin stop, tabadmin cleanup, and then tabadmin start`:

```
tabadmin cleanup --restart
```

The files and database entries that are removed by `tabadmin cleanup` command depend on whether Tableau Server is running or stopped. Therefore, to clean up all possible files and database entries, you should run `tabadmin cleanup` twice: once when Tableau Server is running, and once when it is stopped. Here’s a summary of what’s removed when you run `tabadmin cleanup` with the server running and stopped.

When you run `tabadmin cleanup` with Tableau Server stopped:

- All log files are removed from `ProgramData\Tableau\Tableau Server-data\tabsvc\logs`. *(Log files from `ProgramData\Tableau\Tableau Server\logs` are not removed.)*

- Temporary files are removed from `ProgramData\Tableau\Tableau Server-temp` and `ProgramData\Tableau\Tableau Server-data\tabsvc\temp`.

- No rows for HTTP requests are removed from the `http_requests` table of the Tableau Server PostgreSQL database, because the database is not accessible when the server is stopped.
When you run `tabadmin cleanup` with Tableau Server running:

- Log files older than the log file rotation interval are removed from `ProgramData\Tableau\Tableau Server\data\tabsvc\logs`. The rotation interval is one day. Active logs and log files from `ProgramData\Tableau\Tableau Server\logs` are not removed.

- Temporary files are not removed.

- Files that are in use (that is, locked by the operating system) are not removed.

- Rows for HTTP requests that are older than seven days are removed from the `http_requests` table of the Tableau Server PostgreSQL database.

**Note:** Rows for HTTP requests older than seven days are also removed when you back up Tableau data. For more information, see Back Up Tableau Server Data.

More Information

For more information about the Tableau Server PostgreSQL repository, see Collect Data with the Tableau Server Repository.

For tips on how to automate running the cleanup and backup commands, see Scripting the backup process.

If you have created a log file archive but you no longer need it, you can remove it from the server by using the **Delete Snapshot** option on the Status page. For more information, see Archive Logs on Status Page (Snapshot).

**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 use the Process Status table on the Server Status page to view the state of
Tableau processes on each Tableau server:

![Server Status Table]

Possible status indicators are listed at the bottom of the table:
When Tableau Server is functioning properly, most processes will show as Active, Busy or Passive (Repository):

- **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
- **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 appears below the status icon:

![Ready for removal](image)

For more information about troubleshooting process status, see Troubleshoot Server Processes.

**Access Status Remotely**

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 primary 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:
   ```
   cd "C:\Program Files\Tableau\Tableau Server\10.5\bin"
   ```
2. Enable remote access by typing the following:
   ```
   tabadmin set wgserver.systeminfo.allow_referrer_ips <ip
   ```
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:

```
tabadmin set wgserver.systeminfo.allow_referrer_ips 10.32.139.31
```

If you are enabling remote access for more than one computer, use commas to separate each IP address.

```
tabadmin set wgserver.systeminfo.allow_referrer_ips 10.32.139.31,10.32.139.35
```

3. Commit the configuration change:

```
tabadmin config
```

4. Restart Tableau Server:

```
tabadmin restart
```

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`).

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"/>
      <vizqserver 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="2015-02-27T20:30:48.5642"/>
      <clustercontroller worker="my_tableau_server:12012" status="Active"/>
      <coordination worker="my_tableau_server:12000" status="Active"/>
    </machine>
  </machines>
  <service status="Active"/>
</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, DecommissioningReadOnly, and DecommissionFailedReadOnly

- **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

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:
Possible status indicators are:

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 `tabadmin zip-logs`) in case you need them for Support, then remove unnecessary files (`tabadmin cleanup`).
2. In Windows Task Manager, stop the clustercontroller.exe process tree and let it restart automatically.
3. Restart Tableau Server.
4. Clean up the coordination service (ZooKeeper) files: Stop the cluster (\texttt{tabadmin stop}), clean up files (\texttt{tabadmin cleanup --reset-coordination}), and then start the cluster (\texttt{tabadmin start}).

5. If Cluster Controller continues to show as down, save the log files (\texttt{tabadmin zip-logs}) 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.

**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.

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 (\texttt{tabadmin ziplogs}) in case you need them for Support, and then remove unnecessary files (\texttt{tabadmin cleanup}).
2. Stop the filestore.exe process using Windows Task Manager and let it restart automatically.
3. Restart Tableau Server.
4. Clean up the coordination service (ZooKeeper) files: Stop the cluster (tabadmin stop), clean up files (tabadmin cleanup --reset-coordination), and then start the cluster (tabadmin start).
5. If the file store continues to be down, save the log files (tabadmin 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 (tabadmin 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 tabadmin 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 tabadmin.log file on the primary node and workers for errors.
4. Stop Tableau Server (tabadmin stop) and then try running the tabadmin decommission command again.
5. Put the file store node back into read/write mode (tabadmin 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 occured 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 (tabadmin restart).
3. Check disk space to make sure there is free space. Collect logs (tabadmin zip-logs) in case you need them for Support, and then cleanup files (tabadmin cleanup).
4. Restart Tableau Server.
5. Stop Tableau Server, collect logs and cleanup coordination service files (tabadmin cleanup --reset-coordination)
7. If the repository continues to show as down, collect logs (tabadmin 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.

Archive Logs on Status Page (Snapshot)

You can generate and download a snapshot (archive) of the Tableau Server log files from a web browser, without opening a command prompt. This zipped snapshot contains a copy of up to seven days of log file data from Tableau Server and any worker servers (if you have a distributed environment). The snapshot process does not change or remove either the Tableau Server log files or the log archives created with tabadmin.
**Note** To specify the amount of data you want to collect or the name of the zip file you are creating, use tabadmin to create an archive of server logs. For more information, see Archive Logs on Command Line (tabadmin).

To generate a snapshot of server log files:

1. Open the Status page:
   - Multi-site: Select **Server > Status**.
   - Single-site: Select **Status**.

2. Click **Generate Snapshot** to create a snapshot of the Tableau Server logs. The Generate Snapshot button is available only if there is no existing snapshot.

   **Note**: This option is available whether or not you have created log archives with tabadmin.

3. Select the number of days of logs you want to include. The default is **Last 7 days**, but you might want to select fewer if you want to reduce the size of the zip file. If you are collecting logs after reproducing an issue, include at least three days worth of logs.

4. Click **Download Snapshot** to download the log snapshot to your web browser’s
default download location. This option is available after you create a snapshot.

Google Chrome shows you the download in the bottom of the window:

5. Click the arrow and then click **Open** to unzip the snapshot or **Show in folder** to see where it was downloaded:

6. (Optional) Click **Delete Snapshot** to delete a log snapshot. This option is available after you create a snapshot. You need to delete the existing snapshot before you can create a new one.
For example, you might want to delete the snapshot that you created before an event that you want to investigate.

Uploading log archives for Tableau Support

If you are creating the archive to send to Tableau Support, see the Knowledge Base for information about how to upload large files.

Rebuild the Search Index

If search is returning incomplete or incorrect results, or if the Search & Browse process is down for an extended period of time, you may need to rebuild the search index.

**Important:** The recommended way to reindex search is to use the `tabadmin rein-dex` command while Tableau Server is stopped. Reindexing while the server is running can result in content, including sites and projects, temporarily disappearing from server pages.

The search index is built or rebuilt at key points during installation or upgrade of Tableau Server, when you restore a backup, and when you add the Search & Browse process to a new or existing node. The index is kept updated by a background task when content changes. If necessary you can force a rebuild of the index using the `tabadmin rein-dex` command.
1. To rebuild the search index, click **Status**.

   ![Server Status](image)

   In a multi-site environment, select **Server > Status**.

2. At the bottom of the page, click **Rebuild Search Index**.

   ![Rebuild Search Index](image)

   **Note:** You might not see all available server content while the index is rebuilding, and larger search indexes can take longer times to finish rebuilding. Reindexing first removes all content from the index, and then re-adds the content to the index. If you do this while Tableau Server is running, users who are logged into the server will see content disappear, and then slowly start to reappear in server pages. Reindexing while Tableau Server is stopped provides a better user experience.

**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 site, 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 on a regular schedule or on-demand on the **General** tab of the **Settings** page for the server.
The **Last synchronized** time indicates the time that synchronization most recently began.

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**.

![Minimum Site Role dialog box](image)
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.

Set the Default Start Page for All Users

By default, Tableau Server installs with the Views page as the default start page for all users. As the administrator, you can change this to another page that all users have access to, such as the Workbooks page. Individual users will be able to override your setting (search for "Access Your Profile and Account Settings" in the Tableau Server Help for details).

To set the default start page for all users:

1. Navigate to the page you want to be the default page.
2. Click your name on the upper right corner of the page.
3. Select **Make This the Start Page for All Users**.

Disable Automatic Client Authentication

After Tableau Desktop 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_expiry_in_seconds` and `refresh_token.idle_expiry_in_seconds` options. See `tabadmin` configuration options.

As a Tableau Server administrator 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**.

**Tableau Server Monitor**

Tableau Server Monitor is installed as part of Tableau Server and can be accessed in the Windows system tray.

![Tableau Server Monitor screenshot](image)

Using this tool you can start and stop the server, open Tableau Server, and display server status.
Open the Server

This command launches Tableau Server in your web browser. This is an easy way to access the web application and the associated maintenance tools.

Start/Stop the Server

You can start and stop the server using these commands. When you stop the server you make it unavailable to all of your users and terminate any sessions that are currently in progress. If someone is publishing a workbook when the server is stopped, the process is abandoned. As a result, only some of the worksheets in the workbook may be published to the server. Because stopping the server can be very disruptive to your users, be sure to warn them prior to this operation or plan maintenance during non-business hours.

Restart the Server

This command restarts the server. While the server is restarting it will be unavailable to all users. Be sure to warn your users of the outage prior to this operation. You will need to restart the server if you make changes to the Tableau Server configuration.

Display Status

This command opens a screen tip containing the status of each process. For more detailed status, use the Maintenance page.

Manage Product Keys

This command opens the product key manager where you can add and remove product keys.

Exit

This command closes Tableau Server Monitor. It does not stop Tableau Server. You can re-open the application by selecting All Programs > Tableau Server 10.5 > Tableau Server Monitor on the Windows Start menu.
Server Settings (General)

The following settings are available on the **General** page in **Server - Settings**.

<table>
<thead>
<tr>
<th>Setting</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Embedded Credentials - Allow publishers to embed data source credentials in a workbook</strong></td>
<td>Allows publishers to attach passwords to published workbooks that will automatically authenticate web users to connect to data sources. The passwords are attached to workbooks and are only accessible on server. That is, when the workbook is opened in Tableau Desktop, users will 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.</td>
</tr>
<tr>
<td><strong>Embedded Credentials - Allow publishers to schedule data extract refreshes</strong></td>
<td>Allows publishers to assign workbooks to schedules. This option is only available if <strong>Allow publishers to embed data source credentials in a workbook</strong> is enabled. When this setting is enabled, publishers will see scheduling options in the Publish dialog box.</td>
</tr>
<tr>
<td><strong>Saved Credentials - Allow users to save passwords for data sources</strong></td>
<td>Allows users to choose &quot;Remember my password&quot; and save data source passwords across multiple visits, browsers, and devices. (By default, users can choose to &quot;Remember my password until I sign out,&quot; which lets them save their password only for a single browser session.)</td>
</tr>
<tr>
<td></td>
<td>As an administrator, you can <a href="#">clear all saved passwords</a> at any time. In addition, users can clear their own saved passwords.</td>
</tr>
<tr>
<td>Saved Credentials - Allow users to save OAuth access tokens for data sources</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. For more information, see OAuth Connections.</td>
</tr>
<tr>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td>Connected Devices - Allow devices to 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>Guest Access - Enable Guest account</td>
<td>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. This option can be used with <strong>Enable automatic logon</strong>, an option you can select during Setup.</td>
</tr>
<tr>
<td>Default Start Page</td>
<td>Takes you 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</td>
</tr>
<tr>
<td>Section</td>
<td>Description</td>
</tr>
<tr>
<td>---------</td>
<td>-------------</td>
</tr>
<tr>
<td><strong>Language and Locale</strong></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><strong>Active Directory Synchronization - Synchronize Active Directory groups on a regular schedule</strong></td>
<td>Controls the synchronization of all Active Directory groups in Tableau Server based on a schedule that you specify after you select the option <em>Synchronize Active Directory groups on a regular schedule</em>. For more information, see Synchronize All Active Directory Groups on the Server.</td>
</tr>
</tbody>
</table>
| **Recommendations Trainer** | Suggests server content, such as data sources and tables, to Tableau Desktop users. Content suggestions are based on popularity of the content or on content frequently used by other users who are similar to the current user.  
Scheduling options control 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. New usage information includes information such as "Laura Rodriguez used the Food Catering data source" and "Henry Wilson used the Monthly Sales data source." |
If you notice any 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 "Recommendations Trainer" task in the Background Tasks for Non Extracts view.

If you want to disable Recommendations, change the `tabadmin set` option, "recommendations.enabled", to false.

| Reset to Default Settings | Returns any server settings that have been changed since setup to their original state. |

## Troubleshooting

### 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 to help you resolve an issue.

You can create a zipped log file archive (snapshot) from the command line on the server, or using the Generate Snapshot option on the Maintenance page. The zipped archive contains copies of the logs you can copy or download using a web browser, and send to Tableau Support. Once you have a copy of the archive, you can delete the archive from your server. For more information on creating, downloading and deleting log file archives, see Archive Logs on Status Page (Snapshot).

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 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 Archive Log Files.

   **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 server configuration file** (\config\tabsvc.yml) to get a basic understanding of the server environment.
6. **Review the admin log** (\logs\tabadmin.log) to understand any maintenance that has been done on the server.

   Search for run as: <script> to find entries specific to tabadmin activity.
7. **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 \vizqlserver 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).

8. **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).

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.

See the following topics for more information:

- Quick Start: Generate a Snapshot of Server Logs
- Server Log File Locations
- Change Logging Levels

**Tableau Server Processes**

Tableau Server installs a number of processes that work together to deliver the features that make up Tableau Server.


In this article

- Configuring processes
- Licensed processes
- List of processes

Configuring processes

Certain processes listed below cannot be configured: cluster controller and coordination service are installed on every node as part of the base install. They are required on every server node and do not count against a core-based license. Data engine is installed when you install file store, Vizportal, VizQL Server, data server, or backgrounder, and cannot be installed separately. Every node that has an instance of one of those processes will also have a single instance of the data engine process.

The topics Performance Tuning Examples and High Availability describe some of the approaches you can take when configuring processes. High-level status for each process is displayed on the server’s Status page and more detailed information related to some of the processes—such as the background process—is in the Administrative Views topic.

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 primary Tableau Server computer to verify they are licensed. If they cannot confirm there is a valid license, for example, if the primary 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.

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.
List of processes

For information on log files generated by these processes, see Server Log File Locations.

<table>
<thead>
<tr>
<th>Process</th>
<th>Purpose</th>
<th>Multi-Threaded</th>
<th>Performance Characteristics</th>
<th>Licensed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Application Server</td>
<td>Handles the web application, REST API calls, supports browsing and searching</td>
<td>Yes</td>
<td>Only consumes noticeable resources during infrequent operations, like publishing a workbook with an extract, or generating a static image for a view. Its load can be created by browser-based interaction and by tabcmd.</td>
<td>Yes</td>
</tr>
<tr>
<td>vizportal.exe</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Backgrounder</td>
<td>Executes server tasks, including extract refreshes, subscriptions, ‘Run Now’ tasks, and tasks initiated from tabcmd</td>
<td>No</td>
<td>A single-threaded process where multiple processes can be run on any or all machines in the cluster to expand capacity. The backgrounder normally doesn’t consume much process memory, but it can consume CPU, I/O, or network resources based on the nature of the workload presented to it. For example, performing large extract refreshes can use network bandwidth to retrieve data. CPU resources can be consumed by data retrieval or complex tabcmd tasks.</td>
<td>Yes</td>
</tr>
<tr>
<td>backgrounder.exe</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cache Server</td>
<td>Query cache</td>
<td>No</td>
<td>A query cache distributed and shared across the server cluster. This in-memory cache speeds user exper-</td>
<td>No</td>
</tr>
<tr>
<td>redis-server.exe</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Process</td>
<td>Purpose</td>
<td>Multi-Thraed</td>
<td>Performance Characteristics</td>
<td>Licensed</td>
</tr>
<tr>
<td>------------------------------</td>
<td>-------------------------------------------------</td>
<td>--------------</td>
<td>-----------------------------------------------------------------------------------------------</td>
<td>----------</td>
</tr>
<tr>
<td>Cluster Controller cluster-</td>
<td>Responsible for monitoring various components,</td>
<td>n/a</td>
<td>Included in the base install on every node.</td>
<td>No</td>
</tr>
<tr>
<td>controller.exe</td>
<td>detecting failures, and executing failover when needed</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Coordination Service zooKeeper.exe</td>
<td>In distributed installations, responsible for ensuring</td>
<td>n/a</td>
<td>Always installed on the primary node. For server installations with three to five nodes, also installed on the first two worker nodes. For server installations of more than five nodes, also installed on the first four worker nodes.</td>
<td>No</td>
</tr>
<tr>
<td>Process</td>
<td>Purpose</td>
<td>Multi-Threaded</td>
<td>Performance Characteristics</td>
<td>Licensed</td>
</tr>
<tr>
<td>--------------</td>
<td>-------------------------------------------------------------------------</td>
<td>----------------</td>
<td>----------------------------</td>
<td>----------</td>
</tr>
<tr>
<td>Data Engine</td>
<td>there is a quorum for making decisions during fail-over</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>hyperd.exe</td>
<td>Creates extracts and processes queries.</td>
<td>Yes</td>
<td>The data engine’s workload is generated by requests from the VizQL server, application server, API server, data server, and backgrounder server processes. The data engine services requests from most of the other server processes as well. It is the component that loads extracts into memory and performs queries against them, and writes the extracts to the database. The data engine is multi-threaded to handle multiple requests at a time. For more information about CPU and memory usage, see The New Data Engine in Tableau Server.</td>
<td>Yes</td>
</tr>
<tr>
<td>Process</td>
<td>Purpose</td>
<td>Multi-Threaded</td>
<td>Performance Characteristics</td>
<td>Licensed</td>
</tr>
<tr>
<td>------------------</td>
<td>------------------------------------------------------</td>
<td>----------------</td>
<td>---------------------------------------------------------------------------------------------</td>
<td>----------</td>
</tr>
<tr>
<td>Data Server</td>
<td>Manages connections to Tableau Server data sources</td>
<td>Yes</td>
<td>Because it’s a proxy, it’s normally only bound by network, but it can be bound by CPU with enough simultaneous user sessions. Its load is generated by browser- and Tableau Desktop-based interaction and extract refresh jobs for Tableau Server data sources.</td>
<td>Yes</td>
</tr>
<tr>
<td>File Store</td>
<td>Automatically replicates extracts across data engine nodes</td>
<td>n/a</td>
<td>A data engine is automatically installed when file store is installed, unless the node already has an instance of data engine.</td>
<td>No</td>
</tr>
<tr>
<td>Repository</td>
<td>Tableau Server database, stores workbook and user metadata</td>
<td>n/a</td>
<td>Normally consumes few resources. It can become a bottleneck in rare cases for very large deployments (thousands of users) while performing operations such as viewing all workbooks by user or changing permissions. For more information, see Tableau Server Repository.</td>
<td>No</td>
</tr>
<tr>
<td>Search &amp; Browse</td>
<td>Handles fast search, filter, retrieval</td>
<td>Yes</td>
<td>The process is memory bound first, and I/O bound second. The amount of memory used scales with the amount of content (number of sites/-</td>
<td>No</td>
</tr>
<tr>
<td>Process</td>
<td>Purpose</td>
<td>Multi-Threaded</td>
<td>Performance Characteristics</td>
<td>Licensed</td>
</tr>
<tr>
<td>------------------</td>
<td>----------------------------------------------</td>
<td>----------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>----------</td>
</tr>
<tr>
<td>VizQL Server</td>
<td>Loads and renders views, computes and executes queries</td>
<td>Yes</td>
<td>Consumes noticeable resources during view loading and interactive use from a web browser. Can be CPU bound, I/O bound, or network bound. Process load can only be created by browser-based interaction. Can run out of process memory.</td>
<td>Yes</td>
</tr>
<tr>
<td>vizqlserver.exe</td>
<td>and display of content metadata on the server</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Archive Log Files**

You can create archives (snapshots) of log files in two different ways: from the Status page using a browser, or from a command prompt using `tabadmin` on Tableau Server. Creating a log file archive gives you a zipped snapshot of logs that you can use for troubleshooting or to send to Tableau Support for help with an issue.

**Quick Start: Generate a Snapshot of Server Logs**

Server administrators can quickly generate and download a zipped snapshot of Tableau Server logs from the Server Status page. The snapshot contains a copy of up to seven days of log information and does not affect the actual logs on the server. You can create the snapshot from any browser, and there’s no need to stop the server first.
1 Navigate to the Snapshot Feature

On the **Server > Status** page, scroll to the bottom of the page:

![Server Status Table]

2 Generate a Snapshot

Click **Generate Snapshot**. If you're running a distributed installation of Tableau Server this will collect logs from all servers in the cluster.
You do not have to stop the server before generating a snapshot.

The **Generate Snapshot** button is available only if no snapshot exists. If an earlier snapshot exists, you need to delete it before generating a new snapshot. Download the existing snapshot first, if you think it contains information you might need.

3 Download a Snapshot

Click **Download Snapshot** to copy the zipped log files to your local computer:

The **Download Snapshot** button is available after the snapshot is generated. The downloaded snapshot is saved to the default download location for your web browser.

4 Delete a Snapshot

Click **Delete Snapshot** to remove an existing snapshot from Tableau Server:
Deleting the snapshot does not delete Tableau Server log files. You are just deleting the snapshot created from those files.

Archive Logs on Status Page (Snapshot)

You can generate and download a snapshot (archive) of the Tableau Server log files from a web browser, without opening a command prompt. This zipped snapshot contains a copy of up to seven days of log file data from Tableau Server and any worker servers (if you have a distributed environment). The snapshot process does not change or remove either the Tableau Server log files or the log archives created with tabadmin.

**Note** To specify the amount of data you want to collect or the name of the zip file you are creating, use tabadmin to create an archive of server logs. For more information, see Archive Logs on Command Line (tabadmin).

To generate a snapshot of server log files:

1. Open the Status page:
   - Multi-site: Select **Server > Status**.
   - Single-site: Select **Status**.

2. Click **Generate Snapshot** to create a snapshot of the Tableau Server logs. The Generate Snapshot button is available only if there is no existing snapshot.
Note: This option is available whether or not you have created log archives with tabadmin.

<table>
<thead>
<tr>
<th>Log Files</th>
</tr>
</thead>
<tbody>
<tr>
<td>Date Generated</td>
</tr>
<tr>
<td>None</td>
</tr>
</tbody>
</table>

3. Select the number of days of logs you want to include. The default is Last 7 days, but you might want to select fewer if you want to reduce the size of the zip file. If you are collecting logs after reproducing an issue, include at least three days worth of logs.

4. Click **Download Snapshot** to download the log snapshot to your web browser's default download location. This option is available after you create a snapshot.

Google Chrome shows you the download in the bottom of the window:
5. Click the arrow and then click **Open** to unzip the snapshot or **Show in folder** to see where it was downloaded:

6. (Optional) Click **Delete Snapshot** to delete a log snapshot. This option is available after you create a snapshot. You need to delete the existing snapshot before you can create a new one.
For example, you might want to delete the snapshot that you created before an event that you want to investigate.

Uploading log archives for Tableau Support

If you are creating the archive to send to Tableau Support, see the Knowledge Base for information about how to upload large files.

Archive Logs on Command Line (tabadmin)

If you have command line access on the primary Tableau Server computer, you can archive Tableau Server log files using the `tabadmin ziplogs` command.

This command creates a zip file containing all of the log files and is useful when you’re working with Tableau Support. If you are running a distributed installation of Tableau Server, perform this step from the primary server. Any worker logs will be included in the zip file.

You may also want to create a log file archive before you run the `tabadmin cleanup` command, because that command removes logs. The ziplogs command does not remove the log files, rather it creates an archive by copying them into a zip file. For more information about cleaning up Tableau Server files, see Remove Unneeded Files.

**Note:** The `tabadmin ziplogs` command may generate messages like “zip error: Nothing to do!” These are generally related to specific steps in the zip process and do not mean the log file archive is empty or the entire archive process failed.

To create a log file archive:
1. Open a command prompt as administrator and navigate to the Tableau Server bin directory. For example:

   \texttt{cd "C:\Program Files\Tableau\Tableau Server\10.5\bin"}

2. Create the zip file by typing \texttt{tabadmin ziplogs -l -n <filename>} where \texttt{<filename>} is name of the zipped archive file you want to create. Choose a unique name with no spaces. Tableau will not overwrite an existing file. To create the file in a location other than the \texttt{\bin} directory, include the path with the file name. The path cannot be a UNC path.

   For example:

   \texttt{tabadmin ziplogs -l -n my_logs}

   If you don't specify a file name, the file is named \texttt{logs.zip}.

   You can also use \texttt{-d mm/dd/yyyy} to only include logs generated since a certain date. For example:

   \texttt{tabadmin ziplogs -l -n -d 12/14/2015}

   The above command creates a zipped file named \texttt{logs.zip} that includes logs dated December 14, 2015 up to the present; earlier logs are excluded. The \texttt{-n} option captures information about the server environment, including which ports are in use. To see a list of all the \texttt{ziplogs} options, type \texttt{tabadmin ziplogs -h}.

   You can find the zipped log file archive in the Tableau Server bin directory unless you specified a different location by including a path with the file name.

   **Uploading log archives for Tableau Support**

   If you are creating the archive to send to Tableau Support, see the Knowledge Base for information about how to upload large files.

   **Server Log File Locations**

   By default, Tableau Server log file archives are gathered in a zip file called \texttt{logs.zip} (you can specify a different name if you create the archive using \texttt{tabadmin}). You can copy the
archive from the server to a local computer and open it there, or send it to Tableau Support. When you unzip the archive, a series of folders are created with related log files. This table explains the possible contents of each folder, along with the original location the files came from on the Tableau Server, the process that created the log files, and details about the files.


Log Archive File Locations (unzipped archive)

<table>
<thead>
<tr>
<th>Files/folders 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>build-version.txt</td>
<td>The build version of Tableau Server.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>tabsvc.yml</td>
<td>Tableau Server configuration file.</td>
<td></td>
<td></td>
<td>\config</td>
</tr>
<tr>
<td>assetkey-encryption</td>
<td>Logs related to repository encryption.</td>
<td>assetkey-encryption.log</td>
<td>tabadmin assetkeys</td>
<td>\log-si\assetkeyencryption</td>
</tr>
<tr>
<td>backgrounder</td>
<td>Logs related to subscriptions and scheduled activities like extract</td>
<td>backgrounder-#.log, spawn.#####.log, tomcat-#.#####-##-#.log</td>
<td>backgrounder.exe</td>
<td>\log-s\backgrounder</td>
</tr>
</tbody>
</table>
### cacheserver

Logs related to the Cache Server process.

- **Logs**
- **Related to the Cache Server process.**
- **cacheserver.exe**
- \cacheserver

### cluster-controller

Logs related to the Cluster Controller process.

- **Logs**
- **Related to the Cluster Controller process.**
- **clustercontroller.exe**
- \clustercontroller

### config

Configuration files.

- **Configuration settings.**
- **Connections.yml**
- **Workgroup.yml**
- **Tableau Server Configuration**
- \config
<table>
<thead>
<tr>
<th><strong>data-collector</strong></th>
<th>what you expect.</th>
<th>\logs\datacollector</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>data-engine</strong></td>
<td>There will be a hyper log file for each day with information about data extracts and queries, and responses to VizQL server requests.</td>
<td>hyper_####<em>##</em>##<em>##</em>##.-log</td>
</tr>
<tr>
<td><strong>dataserver</strong></td>
<td>Information about connections to Tableau Server data sources.</td>
<td>dataserver-#.log</td>
</tr>
<tr>
<td><strong>httpd</strong></td>
<td>Apache logs. Look here for authentication entries.</td>
<td>access.####-##-##-##-##log error.log startup.log</td>
</tr>
</tbody>
</table>
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.

<table>
<thead>
<tr>
<th>licensing</th>
<th></th>
<th>\logs\licensing</th>
</tr>
</thead>
</table>
| logs      | This is the location of the logs of most interest and usefulness. Look here after reviewing the configuration | tabadmin.log  
            tabconfig.log  
            tablicsrv.log  
            tabsrvlic.log | \logs       |

- 1038 -
tabadmin.log is never overwritten or truncated so it contains all the details.

notify-tabadmin.log contains errors from tabadmin.log (the errors are also included in tabadmin.log).

tablicsrv.log and tabsrvlic.log are related to licensing.

<table>
<thead>
<tr>
<th></th>
<th>PostgreSQL database logs, including files</th>
<th>tabspawn</th>
<th>\logs\pgsql</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>pgsql</strong></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Process</td>
<td>Description</td>
<td>File</td>
<td>Log Location</td>
</tr>
<tr>
<td>---------</td>
<td>-------------</td>
<td>------</td>
<td>--------------</td>
</tr>
<tr>
<td>repository</td>
<td>related to launching server processes.</td>
<td>postgres.exe</td>
<td>\logs\repository</td>
</tr>
<tr>
<td>service</td>
<td></td>
<td>notify-tabsvc.log, tabsvc.log</td>
<td>\logs\service</td>
</tr>
<tr>
<td>solr</td>
<td>Related to search indexing.</td>
<td></td>
<td>\logs\solr</td>
</tr>
<tr>
<td>svcmonitor</td>
<td></td>
<td></td>
<td>\logs\svcmonitor</td>
</tr>
<tr>
<td>tabadminservice</td>
<td>Related to log archives created using the Generate a Snapshot of Server Log Files option.</td>
<td></td>
<td>\logs\tabadminservice</td>
</tr>
<tr>
<td>tabadmwrk</td>
<td>Server Worker Manager process that is used for auto-discovery of worker servers in a distributed</td>
<td>tabadmwrk.exe</td>
<td>\logs\tabadmwrk</td>
</tr>
<tr>
<td>vizportal</td>
<td>Information related to administrative tasks, workbook and permissions management, authentication, sign-ins, initial view requests, and publishing requests.</td>
<td>vizportal-0.log</td>
<td>vizportal.exe</td>
</tr>
<tr>
<td>---</td>
<td>---</td>
<td>---</td>
<td>---</td>
</tr>
<tr>
<td></td>
<td>REST API requests. Log entries related to REST API requests start with the receipt of the HTTP request, followed by</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
information related to the processing of that request. If the request is completed, these log entries end with the HTTP response code for that request.

| VizQLServer | Related to showing and interacting with views. When running multiple instances of VizQL Server, the instances are dis- | VizQLServer.exe | logs\VizQLServer |
Distinguished by port number.

Notify-production logs contain exceptional events.

Most files are in JSON format. tab-protosrv.txt is created when you open data or connect to data.

backgrounder_####_####_##_##_##_.txt
dataset_####_####_##_##_##_.txt
tabadmin_####_##_##_##_##_.txt
tabprotosrv.txt
tdserver_
vizqlserver_
####_####_##_##_##_.txt

vizqlserver\logs
Tableau Server Log File Locations

If you installed Tableau Server in the default location, the log directory is C:\ProgramData\Tableau\Tableau Server\data\tabsvc\logs. The ProgramData folder is hidden unless you explicitly make it visible.

The following log files track activities related to the web application, database, and index:

C:\ProgramData\Tableau\Tableau Server\data\tabsvc

If you have installed Tableau Server to a non-default location (for example, on a drive other than the system drive), then the logs directory is located at <installation path or drive>\Tableau\Tableau Server\logs. For more information, see Installation directory.

VizQL Logs

These log files track activities related to displaying views, such as querying the database and generating images:

C:\ProgramData\Tableau\Tableau Server-\data\tabsvc\vizqlserver\Logs

Temporary Files

Any file that starts with exe_ in the folder below is a Tableau Server file and can be deleted.
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.

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:** 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.

Change Logging Levels

Set logging levels for Tableau Server using one of several **tabadmin 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>Command</th>
<th>Location of affected logs</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>(path begins with <code>\ProgramData\Tableau\Tableau Server-\data\tabsvc</code>)</td>
</tr>
</tbody>
</table>
Valid process names are backgrounder, vizportal, vizqlserver, or dataserver

For more information, see tabadmin configuration 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 primary server.

To change the logging level:

1. Open a command prompt as administrator and navigate to the Tableau Server bin directory.

   If Tableau Server is installed on the C drive:

   `C:\Program Files\Tableau\Tableau Server\10.5\bin`
   
   or
   
   `C:\Program Files (x86)\Tableau\Tableau Server\10.5\bin`

2. Stop Tableau Server by typing:

   `tabadmin stop`

3. Set the logging level to by typing `tabadmin set [command][option]`

   where [command] is `<process>.native_api.log.level` or `vizqlserver.log.level`
and [option] is a valid logging level.

Examples:

- `tabadmin set backgrounder.native_api.log.level debug`
- `tabadmin set vizqlserver.log.level warn`
- `tabadmin set vizportal.log.level debug`

4. Restart Tableau Server by typing:

    `tabadmin restart`

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:

- `tabadmin set backgrounder.native_api.log.level -d`
- `tabadmin set vizportal.log.level -d`
- `tabadmin set vizqlserver.log.level -d`

Handle an Unlicensed Server

Tableau offers two licensing models: user-based and core-based. 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 it allows. Each user is assigned a unique user name on the server and is required to identify himself 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. You can see your products keys and add new ones by selecting Start > All Programs > Tableau Server > Manage Product Keys.

Unlicensed Core-Based Server

A core-based server can become unlicensed for a variety of reasons. A common problem is that any Tableau Server node has more cores than the license allows. When the server is unlicensed you may not be able to start or administer the server. You can, however, manage your licenses using the tabadmin command line tool. Follow the steps below to see a list of your licenses and number of cores by machine.

1. Open a command prompt as administrator and navigate to the Tableau Server bin directory. For example, on a default install, type the following: `cd C:\Program Files\Tableau\Tableau Server\10.5\bin`
2. Type the following: `tabadmin licenses -p`.

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 a worker machine, from communicating with the primary 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.

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 Data Sources

For users to work with Tableau Server data sources, up to three things need to be in place:

- **Permissions for the data source:** Anyone connecting to a data source must have the Connect and View permissions for it. This also applies to users accessing views that connect to data sources. Anyone publishing and modifying data sources must be licensed to Publish and also have the Write/Save As and Download/Web Save As permissions. See Set Permissions on Individual Content Resources for more information.

  Multidimensional (cube) data sources have to be downloaded and used in Tableau Desktop, so they require Download/Web Save As permission. For more information about cubes in Tableau, see Cube Data Sources.

- **Ability to authenticate to the database:** There are several ways you can connect to data in Tableau and control who has access to what. Basically, whichever entity is connecting to the database must be able to authenticate. The entity could be Tableau Server performing an extract refresh. It could be a Tableau Desktop user connecting to a data source that then connects to a live database. It could also be a Tableau Server user who’s accessing a view that connects to a live database. Refer to Data Security to learn more about your options.

- **Database drivers:** If the person who created and published the data source in Tableau Desktop needed to install additional database drivers, you may need to install them on Tableau Server as well. If you are running a distributed installation of Tableau Server where, for example, the data server process is running on a worker server, any required database drivers must be installed there as well as on the primary
server. Other processes require drivers as well. See Database Drivers for more information.

Data Source Error Messages

Here are some errors that workbook authors and other users may encounter as they work with data sources and views:

**Permission to access this Tableau Server data source denied:** Connecting to a data source requires the Connect permission. See Set Permissions on Individual Content Resources for more information.

**Data source not found:** Someone working with a view may see this error if a data source is removed from Tableau Server or if their Connect to Data page needs to be updated. To update the Connect to Data page in Tableau Desktop, click the Refresh icon:

![Refresh icon](image)

**Unable to connect to this Tableau Server data source:** This error may appear if the connection information for the data source has changed—for example, as a result of the database server name changing. Look at the Data Connection information for the data source and confirm that it has the correct settings.

**Unable to list Tableau Server data sources:** This error may occur if a user is trying to access Tableau Server data sources and there are connectivity issues between Tableau Server and Tableau Desktop.

**Can’t connect with a cube data source:** To use a published multidimensional (cube) data source, you must download the data source and use it in Tableau Desktop. Verify that
you have the **Download/Web Save As** permission for the data source. For more information about cubes in Tableau, see Cube Data Sources.

**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 `tabadmin` option `subscriptions.timeout`.

**Can't subscribe**

If you can see a view on Tableau Server and it has a subscription icon (🔒) in the upper right corner, you can subscribe to it.

Two things need to be in place for you to subscribe to a view: Tableau Server needs to be correctly configured (described in Set Up a Server for 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 on Tableau Server but be unable to subscribe to it. This happens for one of three 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.
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://tableauserver/views/SuperStore/sheet1. To generate a view as a PNG, .png could be added to the end of the URL. For example, http://tableauserver/views/SuperStore/sheet1.png.

In versions 8.1, 8.2, or 8.3, view URLs use this syntax: http://tableauserver/views/SuperStore/sheet1#1. To generate a PNG, add .png before the hash tag. For example: http://tableauserver/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://tableauserver/views/SuperStore/sheet1?:iid=2.

To generate a PNG in version 9.0 and later, add .png before the session ID: http://tableauserver/views/SuperStore/sheet1.png?:iid=2

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 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 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 `j smith` it must be stored in Tableau Server as `j smith`.

**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 Archive Log Files.

**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/

![Tableau Server Configuration](image)

Corrected: 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.

Troubleshooting Mutual SSL Authentication

This topic describes possible mutual (two-way) SSL authentication issues and their causes, the messages that users might see, and possible mitigation for the issues.

Issues

- We couldn’t find a valid client certificate. Contact your Tableau Server administrator.
- Invalid user name or password
  - We couldn’t find your user name in the client certificate. Contact your Tableau Server administrator or sign in using your Tableau Server account.
  - Users unexpectedly see a sign-in dialog box that displays an error message
  - We couldn’t find your user name in the client certificate. Contact your Tableau Server administratorCertificate does not contain a valid Tableau Server user name.
  - The user is signed in using an unexpected user name (LDAP mapping)
The user is signed in as the incorrect user (UPN or CN mapping)

For more information about mutual SSL authentication and LDAP, UPN, and CN user mapping, see the following topics:

- Configure Mutual SSL Authentication for Tableau Server on Windows
- Mapping a Client Certificate to a User During Mutual Authentication

**We couldn't find a valid client certificate. Contact your Tableau Server administrator.**

The client is missing a certificate.

If the client has no client certificate, the user sees this message during authentication:

**We couldn't find a valid client certificate. Contact your Tableau Server administrator.**

To resolve the issue, the user should contact the system administrator to generate a certificate for the client computer.

**Invalid user name or password**

The client doesn't support mutual SSL authentication.

Versions of Tableau Desktop older than version 9.1 do not support mutual SSL authentication. If an older version of Tableau Desktop is used to connect to Tableau Server that is configured for mutual SSL authentication, the following can occur:

- If Tableau Server is configured to use fallback authentication, the client displays a sign-in dialog box and the user can enter a user name and password.
- If the server is not configured to use fallback authentication, the user sees the
following message and cannot connect to the server:

Invalid user name or password

For more information about fallback authentication, see Configure Mutual SSL Authentication for Tableau Server on Windows.

We couldn't find your user name in the client certificate. Contact your Tableau Server administrator or sign in using your Tableau Server account.

Client certificates are not published to Active Directory.

If Tableau Server is configured to use Active Directory for authentication, and if user mapping is set to LDAP, Tableau Server sends the client certificate to Active Directory for authentication. However, if client certificates have not been published to Active Directory, authentication fails and the user sees the following message:

We couldn't find your user name in the client certificate. Contact your Tableau Server administrator or sign in using your Tableau Server account.

To resolve this issue, the system administrator should make sure that client certificates are published to Active Directory. Alternatively, the server should be configured to use a different user mapping (UPN or CN), and the system administrator should be sure that client certificates contain user names in the UPN or CN fields.

Users unexpectedly see a sign-in dialog box that displays an error message

If Tableau Server is configured to use mutual SSL authentication and certificates are available for use with users' computers, a user should not see a sign-in dialog box, because Tableau Server uses the certificate to authenticate the user. However, if the server does not recognize the user name in the certificate, the user sees a sign-in dialog box with an error
message that indicates why the certificate was not used. This can occur when all of the following conditions are true:

- Fallback authentication is enabled.

- If the server is using UPN or CN mapping, the user name in the certificate's UPN or CN field is not recognized. If the server is using LDAP mapping, the certificate is not mapped to the user in Active Directory.

To resolve this issue, the system administrator should do the following, depending on how user mapping is configured on Tableau Server:

- LDAP mapping: Make sure that the certificate is linked to the user, that the certificate is available for use with the user's computer, and that the user is configured as a Tableau Server user.

- UPN or CN mapping: Make sure that the certificate is available for the user's computer, that the user name is in the certificate's UPN or CN field, and that the user name matches the user name on Tableau Server (including domain).

We couldn't find your user name in the client certificate. Contact your Tableau Server administrator

Certificate does not contain a valid Tableau Server user name.

The user name in the UPN or CN fields is missing or invalid

When Tableau Server is configured to use UPN or CN mapping, the server reads the user's name from the UPN or CN field of the certificate and then looks up the user name in Active Directory or in the local repository on Tableau Server. (The specific field that the server reads depends on which mapping—UPN or CN—the server is configured to use.) If the field that is supposed to contain the user name has nothing in it, the user sees the following message:

We couldn't find your user name in the client certificate. Contact your Tableau Server administrator.
If a client certificate contains a user name but Active Directory and Tableau Server don't recognize the user name, the user sees the following message:

Certificate does not contain a valid Tableau Server user name.

This can occur when all of the following conditions are true:

- Tableau Server is configured to use UPN or CN mapping.
- Fallback authentication is not enabled.
- The client certificate has no user name in the UPN or CN field, or the user name in the UPN or CN field does not match a user name in Active Directory or on Tableau Server.

To resolve this issue, the system administrator should make sure that the user's certificate has the correct user name in the UPN or CN fields of the certificate.

The user is signed in using an unexpected user name (LDAP mapping)

When the server is configured to use Active Directory authentication and LDAP mapping, the certificate is linked to a user in Active Directory. If the certificate contains a user name in the UPN or CN field, that user name is ignored.

If the intention is that the user should be signed in with the user name in the UPN or CN fields, the server should be configured to use UPN or CN mapping.

The user is signed in as the incorrect user (UPN or CN mapping)

Under some circumstances, the user name in a UPN or CN field in the client certificate can be ambiguous. The result is that a user is signed in to the incorrect identity.

For more information about the conditions under which this issue can occur, see the section “Address user-name ambiguity in multi-domain organizations” in Mapping a Client Certificate to a User During Mutual Authentication.
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 `ProgramData\Tableau\Tableau Server\data\tabsvc\logs\vizqlserver\vizql-*.log`. To increase the logging level from info to debug, use the `tabadmin` setting `vizqlserver.trustedticket.log_level`.

To test your trusted authentication deployment, see Testing 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. The exact reason for this message is written to the `vizql*.log` files in the following folder:

`ProgramData\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 IP address or host name for the computer sending the POST request must be 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.

- **Value of wgserver.trusted_hosts is properly formatted**

  The list of trusted hosts you provided using the `wgserver.trusted_hosts` setting must be a comma-separated list with a space after each comma. For example, the list should be similar to the following: 192.168.1.101, 192.168.1.102, 192.168.1.103, or bigbox1.example.lan, bixbox2.example.lan, bigbox3.example.lan.

- **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 with a Viewer or Interactor license level. You can see a list of users and their license levels by signing in to Tableau Server as an administrator and clicking the Licensing link on the left side of the page.

- **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`

- **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 [Enable automatic login](#).
If you see a 401 error (or a 302 - Redirect error) after you have deployed Tableau Server 10.5, 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.

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 sub-domain 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 tabadmin set option vizqlserver.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 set to change it to false.

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 `tabadmin set backgrounder.failure_threshold_for_run_prevention` command. For more information, see `tabadmin` configuration options.

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):
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:

<table>
<thead>
<tr>
<th>Data</th>
<th>Analytics</th>
</tr>
</thead>
<tbody>
<tr>
<td>Population</td>
<td>Sales by Region</td>
</tr>
</tbody>
</table>

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.

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.

**Troubleshoot Server Processes**

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:

<table>
<thead>
<tr>
<th>Active</th>
<th>Busy</th>
<th>Passive</th>
<th>Unlicensed</th>
<th>Down</th>
<th>Status unavailable</th>
</tr>
</thead>
</table>

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"](image)

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 `tabadmin zip-logs`) in case you need them for Support, then remove unnecessary files (`tabadmin cleanup`).
2. In Windows Task Manager, stop the clustercontroller.exe process tree and let it restart automatically.
3. Restart Tableau Server.
4. Clean up the coordination service (ZooKeeper) files: Stop the cluster (`tabadmin stop`), clean up files (`tabadmin cleanup --reset-coordination`), and then start the cluster (`tabadmin start`).
5. If Cluster Controller continues to show as down, save the log files (`tabadmin zip-logs`) 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.

**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.

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 (`tabadmin ziplogs`) in case you need them for Support, and then remove unnecessary files (`tabadmin cleanup`).
2. Stop the filestore.exe process using Windows Task Manager and let it restart automatically.
3. Restart Tableau Server.
4. Clean up the coordination service (ZooKeeper) files: Stop the cluster (`tabadmin stop`), clean up files (`tabadmin cleanup --reset-coordination`), and then start the cluster (`tabadmin start`).
5. If the file store continues to be down, save the log files (`tabadmin 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 (`tabadmin 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 `tabadmin 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 `tabadmin.log` file on the primary node and workers for errors.
4. Stop Tableau Server (`tabadmin stop`) and then try running the `tabadmin decommission` command again.
5. Put the file store node back into read/write mode (`tabadmin 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 (tabadmin restart).
3. Check disk space to make sure there is free space. Collect logs (tabadmin zip-logs) in case you need them for Support, and then cleanup files (tabadmin cleanup).
4. Restart Tableau Server.
5. Stop Tableau Server, collect logs and cleanup coordination service files (tabadmin cleanup --reset-coordination)
7. If the repository continues to show as down, collect logs (tabadmin 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.

Troubleshoot Inconsistent Process Status

Disclaimer: This topic includes information about a third-party product. Inclusion of this information is not an endorsement of the product, but is provided as a convenience for our customers. Please note that while we make every effort to keep references to third-party content accurate and up to date, the information we provide here might change without notice as the third-party product changes.

Follow the suggestions to resolve issues with Tableau Server process status. For additional troubleshooting steps based on process status viewed on the Status page, see Troubleshoot Server Processes.

Issue

When Tableau Server is configured with multiple networks cards, the Status page may report inconsistent or inaccurate process status. These potential inconsistencies and inaccuracies can result in other functionality such as alerting or notifications unreliability.

Environment

- Tableau Server 9.0 and higher
- Tableau Server computers with multiple network cards (NICs)

Resolution

To address this problem, you can disable the second NIC, or use the following procedure to assign metrics to each NIC on the computer.
Note: Updating DNS or using local routing in the etc\hosts file to refer to the preferred IP address will not resolve this issue.

Assign metrics for each network interface

A metric is a way to indicate the “cost” of using a network interface. The higher the metric, the more expensive it is to use. By default in Windows, Automatic Metric is enabled, but you can manually assign metrics to each network interface to indicate which network interface is preferred. The lower a metric value the more preferred the interface is.

To manually configure metrics for a network interface:

1. In Control Panel, click **Network and Internet**.
2. Click **Network and Sharing Center**.
3. Click **Change adapter settings**.
4. Right-click on a network interface and click **Properties**.
5. Select **Internet Protocol Version 4(TCP/IPv4)** and click **Properties**.
6. On the General tab, click **Advanced**.
7. On the IP Settings tab, clear **Automatic metric** and enter the metric that you want in the **Interface metric** box.

    The metric indicates the cost of using the interface, so give your preferred interface a lower value than the other interface(s) on the computer.

Repeat the process for any other interfaces, giving them metrics based on their preference. The interface that Tableau Server uses should be the preferred interface and have the lowest value metric. For example, give the preferred network interface a metric of 5 and the secondary interface a value of 10.
Disclaimer: Although we make every effort to ensure links to external websites are accurate, up to date, and relevant, Tableau cannot take responsibility for the accuracy or freshness of pages maintained by external providers. Contact the external site for answers to questions regarding its content.

For more information about Windows and using the metric feature for IP routes, see the following Microsoft documentation:

- https://support.microsoft.com/en-us/kb/299540

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 indexing and processes not fully started can be resolved by restarting Tableau Server in a controlled way. To restart Tableau Server, use the `tabadmin restart` command. This will stop all the processes associated with Tableau Server and then restart them.

3. Clean up files associated with the Coordination Service (ZooKeeper). To clean up Coordination Service files, use the `tabadmin cleanup --reset-coordination` command.
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 `tabadmin 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 `tabadmin restart` command from a command prompt. This will shut down any processes that are running, and restart Tableau Server.

Installing Tableau Server

Install fails due to hardware requirements

Starting with version 9.0, Tableau Server cannot install if the computer you are installing on does not meet the minimum hardware requirements. The requirements apply to both primary server computers and worker computers. For details on minimum hardware requirements, see Minimum Hardware Requirements and Recommendations for Tableau Server.

Install or upgrade generates an error when PostgreSQL ODBC driver does not install correctly

In certain circumstances (when a system reboot is pending, or another program is being installed or updated, the Tableau Server PostgreSQL ODBC driver does not install correctly. When this happens, this message displays:
PostgreSQL ODBC driver (64-bit) version 09.03.0400 did not install properly.

**Note:** The version may be different, depending on what version of Tableau Server you are installing.

If this occurs, follow these steps to correct the issue:

1. Check to see if the driver shows as installed in Control Panel.
2. If the driver is not installed, download it from the Tableau Drivers page and install it.
3. If the driver is installed, uninstall it from Control Panel, restart the computer, download the driver, and install it again.

**Upgrading Tableau Server**

Extract migration is slow

Tableau Server 9.0 introduced a more reliable storage mechanism for data extracts called the File Store. Upgrading from a previous version requires migration of the extracts. This can take a long time (up to several hours) if you have a large number of extracts or extracts that have a lot of data. During migration a message displays:

*Migrating extracts to File Store*
*This process may take up to several hours.*

If the migration progress appears to be stalled or stuck, you can verify that migration is continuing by watching the `tabadmin.log`. An entry is written to this log for each extract that is migrated. You can periodically copy the log and open your copy in a text editor like NotePad to verify that entries are being written to it.
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. Zip and save logs using the `tabadmin 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 `tabadmin cleanup` command. For more information, see Remove Unneeded Files

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 `tabadmin reindex` command.
Activating Tableau Server

Tableau Server license activation fails

In some instances Tableau Server license activation may fail with one of the following error 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.tableausoftware.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
- o.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
Requests to the above domains may be on port 80 or 443.

Verify the date and time

Verify the date and time on the primary 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.

Verify FlexNet Licensing Service has started

If the date and time on the Tableau Server computer are correct, verify that the FlexNet Licensing Service is running on the primary Tableau Server computer.

1. On the primary computer, from the Windows Start menu, open services.msc.

2. In the Services dialog box, verify that the status of FlexNet Licensing Service 64 (64-bit) or FlexNet Licensing Service (32-bit) is Started.

   If FlexNet is not listed as Started, right-click FlexNet Licensing Service and select Start.
If the **Start** option is grayed out, the service may be set to Disabled. To enable the service:

a. Right-click FlexNet Licensing Service and select **Properties**.

b. From the **Startup type** drop-down list, select **Automatic**.

c. Click the **Start** button, and then click **OK**.

**Force the product key to be read again**

1. On the primary Tableau Server computer, sign in as administrator and open a command prompt.

2. Change to the Tableau Server bin directory. By default this is:

   ```
   C:\Program Files\Tableau\Tableau Server\10.5\bin
   ```

3. Type the following commands:

   ```
   tabadmin stop
   lmreread
   tabadmin 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 primary Tableau Server computer, sign in as administrator and open a command prompt.

2. Change to the Tableau Server bin directory. By default this is:

   ```
   C:\Program Files\Tableau\Tableau Server\10.5\bin
   ```

3. Type the following command:
serveractutil -view > LicResults.txt

4. Contact Customer Support ([http://www.tableau.com/support/request](http://www.tableau.com/support/request)) and include the LicResults.txt file that you created.

Troubleshoot Desktop License Reporting

When Tableau Server and Tableau Desktop instances are properly configured, Tableau Desktop license usage information is available in two administrative views, Desktop License Usage and Desktop License Expiration. If you can't see these views, check the requirements below. If you can see the views but there is no data in them, you can troubleshoot by using the Troubleshooting topic in the Desktop Deployment Guide: [Troubleshooting license reporting](http://www.tableau.com/support/request). For details on how to configure Tableau Desktop, see [Configure Desktop License Reporting](http://www.tableau.com/support/request) in the Tableau Desktop Deployment Guide.

Administrative views aren't available

The Desktop License Reporting administrative views are available only to Tableau Server administrators. If you do not see links to the Desktop License Usage and Desktop License Expiration views on the Server Status page (select **Manage All Sites** from the sites menu), verify the following:

- You are signed in as a Tableau Server administrator.
- You are running a version of Tableau Server 10.0 or later, and users are running Tableau Desktop version 10.0 or later. Desktop License Reporting is available beginning with version 10.0 of Tableau Server and Tableau Desktop.
- Tableau Server has Desktop License Reporting enabled. (The feature is disabled by default.) For more information, see [Enable and configure Desktop license reporting](http://www.tableau.com/support/request).

Troubleshoot Disk Space Usage on Tableau Server Nodes

When available disk space on a Tableau Server primary or worker node is low, performance can be degraded. If free space falls too low, Tableau Server may begin to perform
erratically. To monitor free disk space, configure Tableau Server to save disk usage information (this is on by default) and, if desired, enable alerts about low disk space. For more information, see Quick Start: Disk Space Alerts.

**Note:** Disk space monitoring measures free disk space on each server node. Available space may be impacted by programs or processes that are not a part of Tableau Server.

If you find that your Tableau Server installation is running into free disk space limitations, you should take steps to make more space available. This topic suggests some ways you can do that.

**Viewing Disk Usage on Tableau Server Nodes**

When disk space usage monitoring is enabled (this is the default), server administrators can use the Server Disk Space administrative view to see current disk space usage, and one month of usage data on your Tableau Server nodes. Use this view to help you determine whether one of your server nodes is experiencing a jump in space usage, or if space usage has increased over time.

**Cleaning Up Tableau Server-Related Files**

To minimize server space used by Tableau Server, you can clean up unnecessary files.

Use the `tabadmin cleanup` command to remove log files, temporary files, and unneeded entries in the PostgreSQL database. If you want to save the logs before you clean them up, you can make an archive. For more information, see Remove Unneeded Files.

Once you have cleaned up log files and temporary files, you may want to use the administrative views to determine which workbooks and data sources are taking up the most space on your server, and whether any of these is not being used. For more information, see Administrative Views.
Identifying and Cleaning Up Other Files

There are a number of tools, like WinDirStat, you can use for viewing disk usage and doing cleanup.

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 extract 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.

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 which checks the number of query
nodes required for Tableau to generate the workbook and displays 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 tabadmin command line tool to run the following commands:

```
tabadmin set native_api.node_limit_checker_pre_rewrite_disable false
tabadmin config
tabadmin restart
```

**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.

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 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.

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 tabadmin command line tool to run the following commands:

```
tabadmin set native_api.preserve_calculations_on_hyper_refresh_conversion true

tabadmin config
```

**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 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 which checks the number of query nodes required for Tableau to generate the workbook and displays an error when the set limit is exceeded.** The error message may something like: "**Logical Query tree has 2348182 nodes, maximum number allowed is 1000000....**":

      Use tabadmin command line tool to run the following commands:

      ```
      tabadmin set native_api.node_limit_checker_pre_rewrite_disable false
      tabadmin config
      tabadmin restart
      ```
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 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....".

5. 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
   ```

   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. 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 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 tabadmin command line tool to run the following commands:

   ```
   tabadmin set native_api.preserve_calculations_on_hyper_refresh_conversion true
   ```
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. Your will need to test your workbooks indi-
vidually 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 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 tabadmin command line tool to run the following commands:

```
tabadmin set native_api.node_limit_checker_pre_rewrite_disable false
tabadmin config
tabadmin restart
```

**Note:** 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. 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.

**Troubleshoot Run As User**

As discussed in the topic, Run As User, Tableau Server requires administrative-like access to the machine on which it is installed. Therefore, when you update the Run As User in Tableau Server Configuration, a background process will configure permissions on the Tableau computer for that account. However, in some complex deployment scenarios you may need to verify or manually configure the Run As User permissions on the local Tableau Server computer. Use this section to verify how permissions are configured on the machines running Tableau Server in your deployment. This section also includes procedures that describe how to set permissions and configure security policies for the Run As User.

**Required Run As User Account Settings**

The Run As User account needs permissions that allows it to modify files and registry settings. In addition, because the Run As User is used as the security context for the Tableau Server Application Manager service (tabsvc), the account must also be given rights to log on as a service.

These permissions are set automatically when you update the Run As User account in Tableau Server Configuration as described in the topic, Create and Update the Run As User Account.
If you have changed the Run As User account, then we recommend revoking the permissions for the previous account. See Revoke Run As User Account Permissions.

If you have recently changed Run As User or are getting permission errors, use this section to confirm that Tableau Server meets the permission requirements that are detailed here. If you’re running a distributed installation, all Run As User permission configurations must be the same across the primary server and all worker nodes.

**Note:** Do not hide the files created by the Tableau Server installer.

Verify Folder Permissions

The account the Tableau Server service runs under is referred to as the Run As User account. The Run As User account needs permission to specific folder resources on the Windows computer.

This topic provides:

- An accounting of the permissions that are required by the Run As User
- Information about installing Tableau Server in non-default locations.
- How to reapply permissions on an existing Run As User account using the Tableau Configuration Utility

This topic refers to the drive where Windows is installed as the *system drive*. The system drive is equivalent to the Windows environmental variable, \%WINDIR\%. The drive where Tableau Server is installed is referred to as the *install drive*.

<table>
<thead>
<tr>
<th>Resource</th>
<th>System or install drive</th>
<th>File Path</th>
<th>Permissions required</th>
</tr>
</thead>
<tbody>
<tr>
<td>folder</td>
<td>system</td>
<td>SYSTEMROOT:\windows\system32</td>
<td>Read, List folder contents</td>
</tr>
<tr>
<td>execut-</td>
<td>system</td>
<td>Read &amp; execute, List folder contents</td>
<td></td>
</tr>
<tr>
<td>able</td>
<td>SYSTEMROOT:windows\system32\cmd.exe</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Drive root</td>
<td>install</td>
<td>root, for example, Local Disk (C:)</td>
<td></td>
</tr>
<tr>
<td>folder</td>
<td>install</td>
<td>PROGRAMFILES\Tableau\Tableau Server</td>
<td></td>
</tr>
<tr>
<td>folder</td>
<td>install (on system drive)</td>
<td>\ProgramData\Tableau\Tableau Server\</td>
<td></td>
</tr>
<tr>
<td>folder</td>
<td>install (on non-system drive)</td>
<td>\Tableau\Tableau Server\data\</td>
<td></td>
</tr>
</tbody>
</table>

When you update the Run As User account in the Tableau Server Configuration utility, a background process (tabconfig) will configure the folder permissions on the Tableau computer for the Run As User account that you specify.

In this case, where you are installing on the system drive into the default folder (C:\Program Files\Tableau), the configuration of folder permissions will be handled by the tabconfig process when you update the Run As User account in Tableau Server Configuration. You do not need to verify or change any folder permissions for this scenario. If you install Tableau Server onto a different drive, you will need to manually configure some permissions.

Installing on non-system drive or in a different folder

If you have installed Tableau Server on a drive other than the system drive, then you will need to configure the system drive to allow the Run As User additional permissions. The fol-
The following table describes the additional permissions that you need to configure if you install Tableau Server on a drive other than the system drive.

<table>
<thead>
<tr>
<th>Resource</th>
<th>System or install drive</th>
<th>File Path</th>
<th>Permissions required</th>
</tr>
</thead>
<tbody>
<tr>
<td>Drive root</td>
<td>install</td>
<td>root, for example, D:\</td>
<td>Read, List folder contents</td>
</tr>
<tr>
<td>folder</td>
<td>system</td>
<td>SYSTEMROOT:\windows\system32</td>
<td>Read, List folder contents</td>
</tr>
<tr>
<td>executable</td>
<td>system</td>
<td>SYSTEMROOT:\windows\system32\cmd.exe</td>
<td>Read &amp; execute, List folder contents</td>
</tr>
</tbody>
</table>

This procedure describes how to set permissions for the Run As User on a given folder in Windows. Use this procedure to set the permissions specified in the table above. This procedure shows how to modify permissions where the Windows system drive is the C:\ drive and the Tableau install drive is on D:\.

1. On the computer hosting Tableau Server (and on Tableau Worker nodes, if distributed), use Windows Explorer to right-click the Tableau Server install drive, for example Local Disk (D:), and select Properties.

2. In the Local Disk Properties Window, select the Security tab.

3. Click Edit, then Add.

4. In the Select Users, Computers, Service Accounts, or Groups dialog box, type the <domain>\<username> for the Tableau Server Run As User account.
5. Click **Check Names** to resolve the account, then **OK** to confirm.

6. With the Tableau Server Run As User account highlighted, select **List folder contents** and **Read**.

7. On the bottom of the **Security** tab, click **Advanced**.

8. In the Advanced Security Settings for Local Disk (C:) window, click **Change Permissions**.

9. In the Advanced Security Settings for Tableau dialog box, highlight the Run As User account and select the **Replace all child object permissions with inheritable permissions from this object** check box.

10. Click **OK** to apply changes to all subfolders and files - this may take a few minutes.

   **Note:** You may encounter one or more warning messages from Windows when you apply these changes. For example, a Windows Security warning message will appear that warns that changing permissions on the root directory may reduce security of the computer. In all cases, verify that the warnings are acceptable given the context and continue with the procedure.

11. Click **OK** to confirm changes, then click **OK** in the Local Disk (C:) Properties dialog box.

12. Follow the steps above to apply the following permissions:
   - **Read, List folder contents:** `SYSTEMROOT:\windows\system32`
   - **Read & execute, List folder contents:** `SYSTEMROOT:\windows\system32\cmd.exe`

13. Click **OK** to exit.
**Note**: In some cases, Windows will display a Recycle Bin error: "The Recycle Bin is corrupted. Do you want to empty the Recycle Bin for this drive?" Click **Yes**.

Reapplying folder permissions

In some organizations, Group Policy or other system management solutions are used to standardize permissions and accounts on application servers. If your organization runs a such a solution, be sure to configure the system to accommodate the folder permissions required by the Run As User account. If the folder permissions for the Run As User account have been changed, you can use Tableau Configuration Utility to reapply the permissions. See **Reapplying Permissions**.

Verify Registry Permissions

The account the Tableau Server service runs under needs permission to modify the registry on the local machine.

In a multi-node cluster, the registry permissions are only granted on the primary node in the cluster.

When you update the Run As User in Tableau Server Configuration, a background process (tabconfig) will configure the registry permissions on the Tableau computer for the account you specify. It’s unlikely that you will need to apply these permissions manually.

Verify that the Run As User has been granted permissions to the following registry branches. If the account that you have specified as the Run As User is a member of the local administrative group or a member of the Domain Admins security group, then the account will not be displayed on the Permissions page.

- HKEY_CURRENT_USER\Software\Tableau
- HKEY_LOCAL_MACHINE\Software\Tableau
Permissions

Tabconfig will grant Read permission and the following Special permissions to these branches:

- Query Value
- Set Value
- Create Subkey
- Enumerate Subkeys
- Notify
- Write DAC
- Write Owner
- Read Control

To view or edit permissions on registry directories:

1. Open the Registry Editor by entering `regedit` in Windows Run, and then clicking OK.
2. In Registry Editor, navigate to the directory where you want to view or edit permissions. Right-click the directory, and then click Permissions....
3. In Permissions, on the Security tab, select the Run As User account, and then click Advanced.
   If you are adding your Run As User account, then click Add and follow the Windows process for adding a user account to the Security tab. After you have added the account, then select the Run As User account, and then click Advanced
4. In Advanced Security Settings, on the Permissions tab, select the Run As User account, and then click Edit.
5. On the Permission Entry, under Basic permissions, verify that Read and Special permissions are selected. Verify that Only apply these permissions to objects and/or containers within this container is not selected.
6. To view or edit Special permissions, click Show advanced permissions.
7. Under Advanced permissions, verify that the permissions enumerated at the beginning of this topic are selected. Verify that Only apply these permissions to objects and/or containers within this container is not selected.
8. If you have set new permissions, then click OK through the multiple windows to finish.
   If you have viewed permissions and not edited anything, then click Cancel to close all windows.
Verify the Local Security Policy

After you specify a Run As User account in Tableau Server Configuration (as described in the topic, Create and Update the Run As User Account), a background process (tabconfig) will update the local security policy on the computer running Tableau Server. Tabadmin will update the local security policy to give "log on as a service" permissions to the Run As User account. This elevated policy is required because the Run As User is used as the security context for the Tableau Server Application Manager service (tabsvc).

**Note:** If the Run As User account that you specify in Tableau Server Configuration is a member of the local administrators or a domain administrator, then tabadmin may not update the local security policy. Updating the Run As User with an account that is a member of local administrators or domain administrators is not a good security practice. We recommend using a domain User account for the Run As User.

In some cases, you may need to manually set security policy for your Run As User. For example, some organizations run Windows Group Policy that remove "Log on as a service" rights that have been set on user accounts. Or an organization may run a policy that creates a permission conflict by specifying "Deny log on as a service." If your organization does this, then you will need to disable or edit such Group Policies so that your Run As User account is not affected.

The following procedure describes how to configure security policy, **Log on as a service**, manually. You can also use the procedure below to verify that your Run As User is appropriately configured with local security policy rights. For example, you should verify that the Run As User account is not specified on the **Deny log on as a service** policy.

If you are running a distributed installation, then configuration must be the same across the primary and all worker nodes.

To verify or update the local security policy:
1. Select **Start > Control Panel > Administrative Tools > Local Security Policy**.

2. In Local Security Policy, open **Local Policies**, select **User Rights Assignments**.

   **To verify or set Log on as a service policy:**

   a. Right-click **Log on as a service** policy and then click **Properties**.

   b. In **Log on as a service Properties**, click **Add User or Group**.

   c. Type the `<domain><username>` for the Tableau Server Run As User account (for example: **MYCO\tableau_server**), and click **Check Names**.

   d. When the account resolves correctly, it is underlined. Click **OK**.

   **To verify Run As User account is not specified in the Deny log on as a service policy:**
a. Right-click **Deny log on as a service** policy, and then click **Properties**.  
b. In **Deny log on as a service Properties**, verify that the Run As User account is not listed. If it is, remove it. When you are finished, click **OK**.

3. Click **OK** to close the Local Security Settings windows.

**Verify Tableau Service Settings**

Confirm that Tableau services are assigned the correct Log On and Startup values. If you are running a **distributed installation** of Tableau Server, perform these steps on the workers as well as on the primary.

1. Log on as administrator to the computer running Tableau Server.

2. On the Tableau Server computer, select **Start > Control Panel > Administrative Tools > Computer Management > Services and Applications > Services**.

3. Open Services and Applications, then click **Services**. Confirm that the following services have the correct settings:

<table>
<thead>
<tr>
<th>Service Name</th>
<th>Logon Value</th>
<th>Startup Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>FLEXnet Licensing Service</td>
<td>Local System</td>
<td>Automatic</td>
</tr>
<tr>
<td>Secondary Logon</td>
<td>Local System</td>
<td>Automatic</td>
</tr>
<tr>
<td>Tableau Server Application Manager (tabsvc)</td>
<td>&lt;domain&gt;&lt;username&gt; This is the Run As User account. See below. If you have not specified a Run As User account, then Network Service account is used.</td>
<td>Automatic</td>
</tr>
<tr>
<td>Tableau Server License Manager (tablicsrv)</td>
<td>Local Service</td>
<td>Automatic</td>
</tr>
<tr>
<td>The License Manager relies on default Windows folder permissions that are applied to the Local</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Service Name</td>
<td>Logon Value</td>
<td>Startup Value</td>
</tr>
<tr>
<td>---------------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td>---------------</td>
</tr>
<tr>
<td>(Runs on primary node only)</td>
<td>Service. If you are seeing licensing errors in the tabadmin log files, then you may need to modify permissions on the Tableau installation directory. See Verify Folder Permissions for more information.</td>
<td></td>
</tr>
</tbody>
</table>

**Note:** Do not change the default settings on the **Recovery** tab of the **Tableau Server Application Manager Properties** dialog box; leave the settings for failure recovery as **Take No Action**. If you change these settings, Tableau Server will restart after being stopped via the **tabadmin** command or **Stop Tableau Server** command.

Changing the Log On Value

To change the **Log On** value for Tableau Server (tabsvc) to the Run As User account:

1. Select **Start > All Programs > Tableau Server > Stop Tableau Server**.
2. Select **Start > All Programs > Tableau Server > Configure Tableau Server**.
3. On the General tab, enter the domain, user name, and password for Tableau Server's Run As User account.
4. Click **OK**, and then select **Start > All Programs > Tableau Server > Start Tableau Server**.

Revoke Run As User Account Permissions

Changing the Run As User account in Tableau Server Configuration Utility does not remove the permissions from the previous account. Therefore, after you change the Run As User account, we recommend manually revoking permissions from the previous account as a
secure best practice. Follow the procedure below to revoke permissions from the previous Run As User account.

If you have changed your Run As User account and your organization uses a forward proxy solution, then you may need to reconfigure the local LAN settings with the new Run As User account. See Configure a forward proxy server for more information.

You must be logged onto the Tableau Server computer with an administrator account to perform the following procedures.

Remove folder permissions

Tableau Server configures permissions on Windows folders according to the location where you install Tableau Server. Before you remove folder permissions, review Verify Folder Permissions to identify the resources for which you will need to remove permissions. Run the following procedure for each of the resources you have identified:

1. For each resource (drive, folder, executable), right-click the resource, and then click **Properties**.
2. On the resource property page, click the **Security** tab, and then click **Edit** to change permissions.
3. On the Permissions page, select the previous Run As User account and then click **Remove**.
4. Click **OK**.

Remove registry permissions

Remove the previous Run As User account from the following registry locations:

- HKEY_CURRENT_USER\Software\Tableau
- HKEY_LOCAL_MACHINE\Software\Tableau

**Warning:** Editing the Windows registry incorrectly can have harmful effects on your computer.
1. Open the Registry Editor by entering `regedit` in Windows Run, and then clicking OK.
2. For each registry directory, right-click the Tableau folder, and then click Permissions.
3. In the Permissions for Tableau page, select the previous Run As User account, and then click Remove.
4. Click OK.

Remove security policies

Remove the previous Run As User account from the following security policies:

- “Log on as a service” policy
- “Allow log on locally” policy

2. In Local Security Policy, open Local Policies, select User Rights Assignments.
3. For each policy:
   a. Right-click the policy and then select Properties.
   b. On the policy property page, select the previous Run As User account, and then click Remove.
   c. Click OK.

Command Line Utilities

tabadmin

You can perform certain administrative tasks and change Tableau Server configuration settings using the tabadmin command line tool. It installs with Tableau Server by default and cannot be installed on other computers.

**Note:** You should only run tabadmin on the primary Tableau Server node, not on worker nodes.
How to use tabadmin

The first step to using tabadmin is to open a command prompt as an administrator:

Next, navigate to Tableau Server's bin directory by entering the following:

```
cd "C:\Program Files\Tableau\Tableau Server\10.5\bin"
```

You're now ready to enter tabadmin commands.

**Important:** If you are using tabadmin to perform more than one task, you must run tasks one after another (serially), rather than at the same time (in parallel).

Change Tableau Server's Configuration from the Command Line

When you enter a command that changes the server's configuration (a `tabadmin set` command for example), you need to follow a sequence of commands:
1. **Stop the server** before issuing the command.

   *Note:* Using the Windows Task Manager to stop Tableau Server can cause problems with your Tableau Server installation. Instead, you should always use the `tabadmin stop` command to stop Tableau Server.

2. Enter the appropriate command to make the configuration change.

3. Run `tabadmin config` to push the change out to all of the server's configuration files.

4. **Start** Tableau Server again.

**Example**

Change the server's configuration using the `tabadmin set` command:

- `tabadmin stop`
- `tabadmin set [option-name value]`
- `tabadmin config`
- `tabadmin start`

**Display Command Line Help**

Use the `tabadmin` built-in help to get a quick description of a command.

To display help for all `tabadmin` commands enter:

- `tabadmin help commands`

To see help for a specific command, enter `tabadmin help <command>`. For example:

- `tabadmin help set`
tabadmin Commands

**Note:** You should only run tabadmin on the primary Tableau Server node, not on worker nodes.


Here are the commands that can be used with the tabadmin command line tool:

- activate
- administrator
- assetkeys
- autostart
- backup
- cleanup
- clearcache
- configure
- customize
- dbpass
- decommission
- delete_webdataconnector
- exportsite
- failoverprimary
- failoverrepository
- get
- get_openid_redirect_url
- importsite
- importsite_verified
- import_webdataconnector
- licenses
- list_webdataconnectors
- manage_global_credentials
- passwd
- recommission
activate

Activates or returns a Tableau Server license online or offline.

Examples

Activate a license offline:

```bash
tabadmin activate --tlf <file.tlf>
```

Return a license offline:

```bash
tabadmin activate --tlr <file.tlr>
```

Activate a license online:
tabadmin activate --activate --key <license>

Return a license online:

tabadmin activate --return --key <license>

<table>
<thead>
<tr>
<th>Option (short)</th>
<th>Option (long)</th>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>--tlf</td>
<td>FILE</td>
<td>For offline activation. If you are offline during Setup, you are prompted to save a .tlq file, which you submit to Tableau. Tableau sends you a .tlf file. You use this .tlf file to activate Tableau Server.</td>
<td></td>
</tr>
<tr>
<td>--tlr</td>
<td>FILE</td>
<td>For offline return (deactivation). The file you use as the argument is the .tlr file that you receive from Tableau.</td>
<td></td>
</tr>
<tr>
<td>--key</td>
<td>PRODUCTKEY</td>
<td>For online activation or return (deactivation) from the command line.</td>
<td></td>
</tr>
<tr>
<td>--trial</td>
<td></td>
<td>For a trial license.</td>
<td></td>
</tr>
<tr>
<td>--activate</td>
<td></td>
<td>Activate the specified license. Use with --tlf, --asr, --key or --trial.</td>
<td></td>
</tr>
<tr>
<td>--return</td>
<td></td>
<td>Deactivate the specified license. Use with --tlr or --key.</td>
<td></td>
</tr>
</tbody>
</table>

See Also

Activate Tableau Server Offline
administrator

Grants or removes the system administrator capability to the named user. This command does not apply to site administrators.

Examples

Remove the system administrator capability from user hwilson:

```
tabadmin administrator hwilson false
```

Give the system administrator capability to user jsmith:

```
tabadmin administrator jsmith true
```

assetkeys

 Creates a new key to encrypt sensitive information, such as credentials for external databases, stored within the Tableau repository, which is a PostgreSQL database that Tableau Server uses internally. The key you create with this command can contain either a passphrase that you specify or one that’s randomly generated.

Keys are generated with AES symmetric encryption. AES uses a block length of 128 bits. Tableau Server uses a 256-bit key length.

Note: Tableau Server must be running when you issue this command.

If you specify your key’s passphrase, it’s a best practice for it to be at least eight characters long. You should also take character sets into consideration. A strong passphrase should contain characters from at least three of the following character sets:

- Lowercase a-z
- Uppercase A-Z
- Digits 0-9
- Non-alphabetic characters

The new key is encrypted and stored in the following key file: asset_keys.yml (ProgramData\Tableau\Tableau Server\data\tabsvc\config). If the key file is lost or corrupted, you can use the assetkeys --validate command to recreate it.

If you use the assetkeys command then later create and restore a backup file (.tsbak), you will need to run the tabadmin assetkeys --validate command after restoring the backup file. By design, backup files do not contain custom encryption keys—even though some data may be encrypted with them. This protects the encrypted values in case the backup file falls into the wrong hands. When you run tabadmin assetkeys --validate after a backup restore, you are prompted to enter the key's passphrase.

If you run unattended restoration from back up, then use the following procedure to restore without entering a passphrase during the validate operation.

1. Before you restore, make a copy of asset_keys.yml. Save the file outside of the Tableau Server installation path (\Program Files\Tableau Server\).
2. Restore the backup. If the asset_keys.yml file was in the \config directory (\Program Files\Tableau\Tableau Server\10.3\config), then it is removed during the restore process.
3. After the restore process is complete, replace the asset_keys.yml file in the \config directory.
4. Run tabadmin assetkeys --validate.

Validation will succeed because the original asset_keys.yml file is in the \config directory. Embedded passwords will function normally.

Examples

Have Tableau Server generate a key and passphrase for you:

```
tabadmin assetkeys --auto_create
```
Generate a key using a passphrase that you specify. You are prompted to enter a passphrase, which will not be displayed as you type:

```
tabadmin assetkeys --create
```

Use the contents of a file as the passphrase:

```
tabadmin assetkeys --create_from_file C:\test\key\password.txt
```

Confirm that the key file `asset_keys.yml` in ProgramData\Tableau\Tableau Server-data\tabsvc\config is valid and consistent with the metadata in the Tableau Repository:

```
tabadmin assetkeys --validate
```

Recreate the file `asset_keys.yml` which is now corrupted or missing from ProgramData\Tableau\Tableau Server\data\tabsvc\config:

```
tabadmin assetkeys --validate
```

You will be prompted for the passphrase.

<table>
<thead>
<tr>
<th>Option (short)</th>
<th>Option (long)</th>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>--auto_create</td>
<td>[length]</td>
<td></td>
<td>Generates a random passphrase to generate the key. Takes an optional argument for the length of the passphrase. You should record the passphrase and keep it in a safe place, as it will be required by --validate if assetkeys.yml is lost or corrupted.</td>
</tr>
<tr>
<td>--create</td>
<td></td>
<td></td>
<td>Generates a key using the passphrase you provide. You are prompted for the passphrase and it will not display as you type it. Your passphrase should be at least 10 characters long and not based on words found in the dictionary.</td>
</tr>
<tr>
<td>--create_from_</td>
<td>FILE</td>
<td></td>
<td>Generates a key using the contents of a file that you provide as the passphrase.</td>
</tr>
<tr>
<td>Option (short)</td>
<td>Option (long)</td>
<td>Argument</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>---------------</td>
<td>----------</td>
<td>-------------</td>
</tr>
<tr>
<td></td>
<td></td>
<td>file</td>
<td></td>
</tr>
<tr>
<td>--validate</td>
<td></td>
<td></td>
<td>Confirms that all asset keys being used internally by Tableau Server are up-to-date. If you lose the asset_keys.yml file (for example, due to file corruption), you can use the --validate option to recreate it. You are prompted for and must enter the passphrase that was used to generate the current asset keys in order to successfully recreate the key file.</td>
</tr>
</tbody>
</table>

See Also

Security

autostart

Specifies whether Tableau Server starts at system start-up time. By default, Tableau Server starts when the computer on which it's installed starts. If autostart is set to off, the tabsvc service will not start (other processes will start) and you will need to explicitly start Tableau Server either using tabadmin start or the Start menu.

After changing the autostart value, you need to run a tabadmin config command to update configuration files on all Tableau Server nodes.

tabadmin autostart [on | off]

Example

Display Tableau Server’s auto-start status:

tabadmin autostart
Start Tableau Server when the operating system starts:

tabadmin autostart on

Do not start Tableau Server when the operating system starts:

tabadmin autostart off

backup

Creates a backup of the data managed by Tableau Server. This data includes Tableau's own PostgreSQL database, which contains workbook and user metadata, extract (.tde or .hyper) files, and configuration data. If you have imported web data connectors using the import_webdataconnector command, the backup process saves copies of the connectors as well. You do not need to stop Tableau Server before you create a backup file.

By default, the backup file is saved in the Tableau Server \bin directory, where you run the tabadmin backup command. You can specify a particular location by including a full path with the backup file name. You can also use the --userdir option to put the backup file into a known location.

Note: The command adds the .tsbak extension to the file name that you specify unless the name already contains that extension.

Examples

Create a backup file in the current directory named tabserv.tsbak:

```bash
tabadmin backup tabserv.tsbak
```

Create a backup file in the C:\backups\tableau folder named tabserv.tsbak:

```bash
tabadmin backup C:\backups\tableau\tabserv.tsbak
```
Append the current date to the backup file name and put temporary files created during the backup process in C:\mytemp\tableau. The backup file **tabserv.tsbak** is created in the directory where you are running the command from:

```
tabadmin backup tabserv.tsbak -d -t C:\mytemp\tableau
```

<table>
<thead>
<tr>
<th>Option (short)</th>
<th>Option (long)</th>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-d</td>
<td>--date</td>
<td></td>
<td>Appends the current date to the backup file name. Date is based on Coordinated Universal Time (UTC).</td>
</tr>
<tr>
<td>-u</td>
<td>--userdir</td>
<td></td>
<td>Places the backup file in the ProgramData\Tableau\Tableau Server folder.</td>
</tr>
<tr>
<td>-t</td>
<td>--tempdir</td>
<td>PATH</td>
<td>Specifies the location for temporary files created during the backup or when verifying database integrity. If the path includes a space, use quotes around the full path.</td>
</tr>
<tr>
<td>-v</td>
<td>--verify</td>
<td></td>
<td>Verifies the integrity of the database. Available beginning with version 9.3.</td>
</tr>
<tr>
<td>--skip-http-truncate</td>
<td></td>
<td></td>
<td>Creates a backup without cleaning entries from the http_requests table in the PostgreSQL database. By default doing a backup will remove all but the most recent 7 days of data in the http_requests table. Use this option to create a backup without cleaning the http_requests table. Available beginning with version 10.3.</td>
</tr>
</tbody>
</table>

**Note:** Tableau recommends that as a best practice you include cleaning the http_requests table as part of your regular database maintenance. For more information, see cleanup and Remove Unneeded Files.
<table>
<thead>
<tr>
<th>Option (short)</th>
<th>Option (long)</th>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>--no-compression</td>
<td></td>
<td>Creates a backup file without using compression. This results in a larger backup file but can reduce the amount of time it takes to complete the backup. Available beginning with version 10.5.8</td>
</tr>
<tr>
<td></td>
<td>--no-config</td>
<td></td>
<td>Creates a backup file without configuration data. The secrets storage configuration bundle is not included in the backup file.</td>
</tr>
</tbody>
</table>

See Also

Back Up Tableau Server Data

cleanup

Reduces the disk space consumed by Tableau Server. Running `tabadmin cleanup` removes log files, temporary files, and rows older than seven days from the `http_requests` table of the Tableau Server PostgreSQL database. If Tableau Server is installed on multiple computers in a cluster, the command can also reset the information maintained by the coordination server that is used to synchronize between nodes and to manage fail-over.

The effect of the `cleanup` command depends on whether the server is running or stopped. For more information, see Remove Unneeded Files.

Examples

Remove log files, temporary files, and HTTP request entries in the PostreSQL database:

tabadmin cleanup
Remove log files and temporary files (leave HTTP request entries in the database untouched):

```
tabadmin cleanup --restart
```

<table>
<thead>
<tr>
<th>Option (short)</th>
<th>Option (long)</th>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-r</td>
<td>--restart</td>
<td></td>
<td>Stops Tableau Server, runs the cleanup command, and starts the server again.</td>
</tr>
<tr>
<td></td>
<td>--reset-coordination</td>
<td></td>
<td>In addition to performing a normal cleanup, removes log files, transaction logs, and snapshots that are maintained by the Tableau Server coordination service (zookeeper) when Tableau Server is running on multiple computers in a cluster. Note that using this option completely resets the coordination service, meaning all state maintained by the coordination service is removed. This option also does the equivalent of a tabadmin con-</td>
</tr>
<tr>
<td>Option (short)</td>
<td>Option (long)</td>
<td>Argument</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>---------------</td>
<td>----------</td>
<td>-------------</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>figure command. For guidelines about when to reset the coordination service, see Troubleshoot Server Processes.</td>
</tr>
</tbody>
</table>

See Also

Remove Unneeded Files

clearcache

Clears the information being cached by the Cache Server process (redis-server.exe). The cache stores information used to render views in order to help speed rendering. Clearing the cache is useful if metadata about views or data sources that might be cached has changed, and those changes should take effect before the resource is removed from the cache in the normal course of server processing. For example, clearing the cache can be useful if you change permissions on a workbook or view and it's important that the changed permissions take effect immediately.

You must stop the server before you run this command.

Examples

tabadmin clearcache

See Also

Tableau Server Processes
configure

Updates Tableau Server’s configuration by forcing an update to all the files in ProgramData\Tableau\Tableau Server\data\tabsvc\<area>. This update includes refreshing the master service configuration file, workgroup.yml (ProgramData\Tableau\Tableau Server\data\tabsvc\config). When you make a configuration change, it’s a best practice to run `tabadmin configure` (or `tabadmin config`) to ensure that all files affecting the server’s configuration are completely updated.

If you are running Tableau Server in a distributed environment and if you have imported web data connectors using the `import_webdataconnector` command or deleted them using the `delete_webdataconnector` command, the `configure` command makes sure that any web data connectors are correctly distributed (imported or deleted) in all nodes where the gateway process is running.

**Examples**

`tabadmin configure`

`tabadmin config`

**See Also**

Reconfigure the Server

`set`

`tabadmin configuration options`
Customize

Customizes the server name that's displayed in tooltips and messages, and the logos that are used by Tableau Server. Note that even if you use this command, the bottom of every server page lists Tableau's copyright information.

Image files you use for logos can be in GIF, JPEG, or PNG format.

Examples

Name

Change the product name used in tooltips from "Tableau Server" to "My Company":

```bash
tabadmin customize name "My Company"
```

Reset the product name to the default:

```bash
tabadmin customize name -d
```

Header logo

Customize the main server header logo. The image can be up to 160 by 160 pixels, but not smaller than 32 by 32 pixels. For best results use an image that's 125 by 35 pixels. If the image is larger than 160 by 160 pixels, it is clipped.

```bash
tabadmin customize header_logo "C:\My Pictures\example.png"
```

Reset the header logo to the default:

```bash
tabadmin customize header_logo -d
```

Sign-in logo

Customize the sign-in page logo. The image can be up to 3000 by 3000 pixels.

```bash
tabadmin customize sign_in_logo "C:\My Pictures\example.png"
```

Reset the sign-in logo to the default:
tabadmin customize sign_in_logo -d

**Logo**

Set the main server header and the sign-in page logo to the same image. The image can be up to 160 by 160 pixels, but not smaller than 32 by 32 pixels. If the image is larger than 160 by 160 pixels, it is clipped.

```
tabadmin customize logo "C:\My Pictures\example.png"
```

Reset the logo to the default:

```
tabadmin customize logo -d
```

<table>
<thead>
<tr>
<th>Option (short)</th>
<th>Option (long)</th>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-d</td>
<td>--default</td>
<td>name</td>
<td>header_logo</td>
</tr>
<tr>
<td>name</td>
<td></td>
<td>NAME</td>
<td>Sets the name to the value in the argument. The default is &quot;Tableau Server&quot;.</td>
</tr>
<tr>
<td>logo</td>
<td></td>
<td>FILE</td>
<td>Sets both the header logo and the sign-in page logo to the image referenced in the file path. If the image is larger than 160 by 160 pixels, it is clipped. If the path includes a space, use quotes around the full path.</td>
</tr>
<tr>
<td>header_logo</td>
<td></td>
<td>FILE</td>
<td>Sets the logo to the image referenced in the file path. For best results use an image that's 125 by 35 pixels. If the image is larger than 160 by 160 pixels, it is clipped. If the path includes a space, use quotes</td>
</tr>
</tbody>
</table>
### Option (short) | Option (long) | Argument | Description
--- | --- | --- | ---
 |  |  |  | around the full path.

| sign_in_logo | FILE | Sets the logo to the image referenced in the file path. The image can be up to 3000 by 3000 pixels. If the path includes a space, use quotes around the full path.

#### See Also

**Change the Name or Logo**

**dbpass**

Enables external access to Tableau's PostgreSQL database (the repository). After you use the dbpass command to allow access to the database, you can connect to and query it using Tableau Desktop to create your own administrative views.

`tabadmin dbpass [--disable] [--username <username>] [password]`

**Note:** The `--username` option is valid starting with Tableau Server 8.2.5. In earlier versions dbpass only enabled the "tableau" user and you could not specify the user. 8.2.5 added a second user called "readonly" and introduced the ability to specify the user you are enabling access for.

#### Examples

Enable access for the `tableau` user and set the password to `p@ssword`:

`tabadmin dbpass p@ssword`
Enable access for the `readonly` user and set the password to `p@ssword`:

```
tabadmin dbpass --username readonly p@ssword
```

Disable external access for the default (`tableau`) user:

```
tabadmin dbpass --disable
```

or

```
tabadmin dbpass --disable --username tableau
```

Disable external access for the `readonly` user:

```
tabadmin dbpass --disable --username readonly
```

<table>
<thead>
<tr>
<th>Option (long)</th>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--disable</code></td>
<td></td>
<td>Disable external access to Tableau's PostgreSQL database for the default remote user (<code>tableau</code>) or, starting in 8.2.5, if a user name is specified, disable remote access for that user.</td>
</tr>
<tr>
<td><code>--username</code></td>
<td><code>tableau</code> or <code>readonly</code></td>
<td>Change the password for the specified user, or, if used with the <code>--disable</code> option, disable access for the specified user. Options for users are <code>tableau</code> and <code>readonly</code>. This option is valid in Tableau Server 8.2.5 or higher.</td>
</tr>
<tr>
<td><code>password provided by user</code></td>
<td></td>
<td>Enable remote access to Tableau's PostgreSQL database for the default remote user (<code>tableau</code>) or, starting in 8.2.5, if a user name is specified, enable access for that user with the given password.</td>
</tr>
</tbody>
</table>

See Also

Collect Data with the Tableau Server Repository
decommission

Prepares Tableau Server File Store nodes for removal from the distributed installation. This command puts the specified nodes into read-only mode so new content cannot be added to the File Store, and makes sure that all content on the node also exists on another File Store node. This command can be run while Tableau Server is running.

**Note:** Remove a decommissioned File Store node before restarting Tableau Server. Restarting automatically re-activates any decommissioned File Store nodes.

```
tabadmin decommission <node1 node2 ...>
```

**Examples**

Decommission worker2:
```
tabadmin decommission worker2
```

Decommission two nodes by IP address:
```
tabadmin decommission 10.32.139.30 10.32.139.22
```

<table>
<thead>
<tr>
<th>Option (long)</th>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>&lt;node1 node2 node 3...&gt;</td>
<td>List of File Store nodes (servers) to decommission. Separate multiple nodes with a space.</td>
</tr>
</tbody>
</table>

**See Also**

Distributed Environments

Maintain a Distributed Environment
delete_webdataconnector

Removes the specified web data connector from the server, or removes all web data connectors. If the web data connector is installed on a cluster, this command removes the specified connector or all connectors from all computers in the cluster.

**Note:** If the server is running in a distributed environment and the delete process is partially successful, users can still access the connector. For more information, see Web Data Connectors in Tableau Server.

**Examples**

```
tabadmin delete_webdataconnector connector1.html
```

```
tabadmin delete_webdataconnector --all
```

<table>
<thead>
<tr>
<th>Option (short)</th>
<th>Option (long)</th>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>--all</td>
<td>--all</td>
<td></td>
<td>Removes all web data connectors from Tableau Server. When you use this option, you do not specify a connector name.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>If the server is configured as a cluster, the command removes all connectors from all the nodes where they are installed.</td>
</tr>
</tbody>
</table>

See Also

import_webdataconnector

list_webdataconnectors

Web Data Connectors in Tableau Server
exportsite

Exports a Tableau Server site, including its users, workbooks, projects, extracts, and data connections, and places it in a file with a .zip file extension. You can then use the exported site file to provision a new site by using the importsite and importsite_verified commands.

Tableau Server must be running when you use the exportsite command. During the export process, Tableau Server locks the site you’re exporting.

Notes: When you import a site that you exported earlier, each user and schedule that is being imported must match an existing user and schedule. For suggestions about how to manage the export and import process to match users and schedules, see Tips for importing to a target with fewer users or schedules than the source site.

If your source site has workbooks that use published data sources, the target site name must match the source site name. The data connections for the workbooks will continue to refer to the source site name and can’t be updated on the new site.

Examples

tagadmin exportsite <site ID> --file <PATH>

or

tagadmin exportsite <site ID> --file <FILE>

Export a site with a site ID of finance to a file named finance_export.zip and place it in Program Files\Tableau\Tableau Server\10.5\bin:

tagadmin exportsite finance --file finance_export
Export the Default site. The site ID for the Default site is "" (double quotations, no space).

`tabadmin exportsite "" --file finance_export`

If you are using Windows PowerShell to run the command, enclose the double quotation marks for the Default site within single quotation marks ('"''). For example:

`tabadmin exportsite '""' --file finance_export`

Export the Default site to a file named `finance_export.zip` and place it in C:\temp\exported sites instead of in the Tableau Server bin directory. Because the path contains a space, it's contained by quotation marks:

`tabadmin exportsite "" --file "C:\temp\exported sites\finance_export"`

Export the site whose site ID is `finance`, name the export site file `financesite.zip`, place the file in C:\sites\exported, and write temporary run-time files to C:\temp_files:

`tabadmin exportsite finance --file C:\sites\exported\financesite --tempdir C:\temp_files`

<table>
<thead>
<tr>
<th>Option (short)</th>
<th>Option (long)</th>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>--file</td>
<td></td>
<td>FILE or PATH</td>
<td>The name or name and location (path) of the exported site file to be created. If you don't specify a path, Tableau Server's bin directory is the assumed location (Program Files\Tableau\Tableau Server\10.5\bin). If you specify a path and the path includes a space, use quotes around the full path.</td>
</tr>
<tr>
<td>--tempdir</td>
<td></td>
<td></td>
<td>The location of temporary files created during export. Use this option if you don't have write access to the Tableau Server installation directory. This option does not determine where the export site file is created.</td>
</tr>
</tbody>
</table>
Option | Option | Argument | Description
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>short</td>
<td>long</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>ated.</td>
<td></td>
</tr>
</tbody>
</table>

See Also

Export or Import a Site

failoverprimary

Identifies a second installation of the primary Tableau Server as the backup primary, or if the primary has failed, identify the backup primary as the new primary and the former primary as the new backup.

**Note:** If you run this command without providing an option, the current computer is assumed to be the primary and no backup primary is identified.

Example

```
tabadmin failoverprimary --primary "<computer name(s) or IPv4 address(es)>"
```

The following command specifies the primary Tableau Server computer (10.32.139.22) and the backup primary (10.32.139.50):

```
tabadmin failoverprimary --primary "10.32.139.22,10.32.139.50"
```
<table>
<thead>
<tr>
<th>Option (short)</th>
<th>Option (long)</th>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>primary</td>
<td>name(s) or IPv4 address(es)</td>
<td>the cluster’s primary.</td>
<td></td>
</tr>
</tbody>
</table>

**See Also**

**High Availability**

Configure for Failover and Multiple Gateways

Use a Backup Primary

**failoverrepository**

Manually identifies a second, passive installation of the PostGRES repository as the active repository.

If Tableau Server is configured for high availability, failover of the repository is automatic. Use the failoverrepository command to manually fail over the repository (for example, if Tableau Server is configured for manual repository failover using the `tabadmin set clustercontroller.pgsqldata.failover false` command).

Tableau Server must be running when you run the failoverrepository command.

```
tabadmin failoverrepository --target <computer name or IPv4 address> | --preferred
```

**Example**

```
tabadmin failoverrepository --target worker_server2
```
Note: This command is persistent. The failover repository remains the active repository until you issue the command again. If you have a preferred active repository configured, use the --preferred option to switch back to that repository.

<table>
<thead>
<tr>
<th>Option (short)</th>
<th>Option (long)</th>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>--target</td>
<td>--target</td>
<td>Computer name or IPv4 address</td>
<td>The Tableau Server repository node to fail-over to.</td>
</tr>
<tr>
<td>--prefered</td>
<td>--preferred</td>
<td></td>
<td>Failover to the repository node that is specified as the preferred active repository.</td>
</tr>
</tbody>
</table>

See Also

High Availability

Configure for Failover and Multiple Gateways

Use a Backup Primary

get

Get the value of Tableau Server configuration options.

Example

tabadmin get [option-name]

Get the backgrounder query limit:

tabadmin get backgrounder.querylimit
get_openid_redirect_url

If Tableau Server is configured to use OpenID Connect for authentication, gets the URL that is used to redirect users from the identity provider (IdP) to Tableau Server after a successful sign-in.

Example

tabadmin get_openid_redirect_url

See Also

OpenID Connect
Configure Tableau Server for OpenID Connect

importsite

Imports a site into Tableau Server. The importsite command is the first of two commands you use to import a site into Tableau Server. To run this command, you need the following:

- **An exported site file.** Tableau Server administrators create this file using the exportsite command. If you have a site on Tableau Online and you want to import it into your own installation of Tableau Server, request an exported site file from Tableau Customer Support.

- **The site ID for the target site.** The target site is the Tableau Server site into which you want to import. The target site must already exist when you run the importsite command; you can't create it as part of the command. The site ID for Tableau
Server's default site is " " (double quotes, no space).

The contents of the site that you import will replace (not amend) the contents of the target site. For example, if your target site has a workbook named MyDashboard.twbx and the site you are importing does not have this workbook, the import process will remove MyDashboard.twbx from the target site.

When you run the importsite command, the command creates a temporary directory containing mapping files in comma-separated-value (CSV) format that define how the exported site's assets (users, workbooks, projects, extracts, and data sources) will be mapped when the site has been imported. It is important that you verify these details. Use a text editor or Microsoft Excel to open the mapping files and make any changes. Any entries with ??? (question marks) represent mappings that couldn't be handled and must be edited. After you verify the mappings, finish the import process using the importsite_verified command.

**Note:** When you import a site that you exported earlier, each user and schedule that is being imported must match an existing user and schedule. For suggestions about how to manage the export and import process to match users and schedules, see Tips for importing to a target with fewer users or schedules than the source site.

**Examples**

```
    tabadmin importsite <site ID> --file <PATH>
```

or

```
    tabadmin importsite <site ID> --file <FILE>
```

Import the file sales_site.zip located in C:\tableau\exported to a site whose site ID is wsales:

```
    tabadmin importsite wsales --file C:\tableau\exported\sales_site.zip
```
Import the file **sales_site.zip**, which is located in C:\Program Files\Tableau\Tableau Server\10.5\bin, to the Default site. The site ID for the Default site is "" (double quotes, no space).

```
tabadmin importsite "" --file sales_site.zip
```

The mapping files for you to verify are placed in ProgramData\Tableau\Tableau Server-data\tabsvc\temp\import_<site ID>_<datetime>mappings. To specify a different directory, use the **--tempdir** option.

Place the files to be verified in C:\temp\site_to_import:

Skip the verification step (not recommended):

```
tabadmin importsite wsales --file "C:\tableau\exported\sales_site.zip" -no-verify
```

<table>
<thead>
<tr>
<th>Option (short)</th>
<th>Option (long)</th>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>--file</td>
<td></td>
<td>PATH</td>
<td>The name and location of the exported site file you are importing. If you don’t specify a path, Tableau Server’s bin directory is the assumed location (Program Files\Tableau\Tableau Server\10.5\bin).</td>
</tr>
<tr>
<td>--no-verify</td>
<td></td>
<td></td>
<td>Skips the verification step and imports the exported site file directly to its new location in your Tableau Server installation. If you choose this option, you do not need to use the <strong>importsite_verified</strong> command.</td>
</tr>
<tr>
<td>Option (short)</td>
<td>Option (long)</td>
<td>Argument</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>--------------</td>
<td>----------</td>
<td>-------------</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td><strong>Note:</strong> Importing a site without verifying the mappings is not recommended.</td>
</tr>
<tr>
<td></td>
<td>--continue-on-ignorable-errors</td>
<td></td>
<td>Ignores any non-fatal errors. Must be used along with --no-verify or with the tabadmin import-site_verified command.</td>
</tr>
<tr>
<td></td>
<td>--tempdir</td>
<td>PATH</td>
<td>The directory where you will verify that the site files have the correct mappings. If you don't specify this option, files are placed in a directory under ProgramData\Tableau\Tableau Server\data\tabsvc\temp.</td>
</tr>
<tr>
<td></td>
<td>--overrideScheduleMapper</td>
<td>PATH</td>
<td>Use the specified mapping file to override the default mapping by name.</td>
</tr>
</tbody>
</table>

See Also

Export or Import a Site
importsite_verified

Performs the second part of an import process for a site on Tableau Server. Before you can use `importsite_verified`, you must first use `importsite`.

The `importsite_verified` command reads from a directory containing CSV files that you have verified, and imports a new site into Tableau Server based on how the site’s assets are mapped in the CSV files. The site that receives the import (the target site) must already exist on Tableau Server.

During the import process, Tableau Server locks the site receiving the import.

Examples

tabadmin importsite_verified <target site ID> --importjobdir <PATH>

Import files from the directory C:\temp\site_to_import to the site whose site ID is **esale**:

<table>
<thead>
<tr>
<th>Option (short)</th>
<th>Option (long)</th>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>--continue-on-ignor-</td>
<td></td>
<td>Ignores any non-fatal errors.</td>
</tr>
<tr>
<td></td>
<td>able-errors</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>--importjobdir</td>
<td>PATH</td>
<td>The directory containing CSV files whose mappings you have verified.</td>
</tr>
</tbody>
</table>

See Also

Export or Import a Site
import_webdataconnector

Installs a web data connector on the server. Users who create workbooks can then reference the web data connector as a data source.

**Note:** Starting with version 10.0 of Tableau Server, the recommended way to make web data connectors available on Tableau Server is to add them to a safe list. For more information, see tabadmin Commands.

**Important:** Before you import a web data connector, make sure that the JavaScript code in the connector does not implement any functionality that should not be on your server.

When the `import_webdataconnector` command finishes importing the connector, the command displays the server URL of the connector. When users want to reference the web data connector as a data source, they need to know this URL. (You can also view the URLs of connectors on your server by using the `list_webdataconnectors` command.)

If the web data connector includes references to an external file, such as to a .css file or .js file, you must make sure that the external file is available from the server, either over the web or as a local file. If the connector references a local file, the local file must be in the same folder as the connector’s .html file relative paths to subdirectories are not supported for imported web connectors. (Make sure that the `<link>` or `<script>` element in the connector correctly references the file as a peer of the connector file.) If the external file is local, you must use the `import_webdataconnector` command to import the external file separately.

If the server includes multiple computers in a cluster, the web data connector is imported to each computer where a gateway process is running.

**Examples**
tabadmin import_webdataconnector connector1.html

tabadmin import_webdataconnector c:\web-dataconnectors\connector1.html --overwrite

tabadmin import_webdataconnector \\my-share\webdataconnectors\connector2.html --overwrite

tabadmin import_webdataconnector connector1.css

**Note:** The connector name can contain only these characters: a-zA-Z0-9 ()._.

<table>
<thead>
<tr>
<th>Option (short)</th>
<th>Option (long)</th>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>--overwrite</td>
<td>--overwrite</td>
<td></td>
<td>Overwrites any existing file on the server that has the same name as the file that you are importing.</td>
</tr>
</tbody>
</table>

See Also

delete_webdataconnector

list_webdataconnectors

Web Data Connectors in Tableau Server

licenses

Displays license information for Tableau Server.

**Examples**

tabadmin licenses
tabadmin licenses -p

<table>
<thead>
<tr>
<th>Option (short)</th>
<th>Option (long)</th>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-p</td>
<td>--processor_</td>
<td></td>
<td>Display the physical core count for the current machine.</td>
</tr>
<tr>
<td></td>
<td>cores</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

list_webdataconnectors

Displays the names or URLs of web data connectors that are installed on the server.

Examples

List the names of the web data connectors.

```bash
tabadmin list_webdataconnectors
```

List the URLs of the web data connectors.

```bash
tabadmin list_webdataconnectors --urls
```

<table>
<thead>
<tr>
<th>Option (short)</th>
<th>Option (long)</th>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>--urls</td>
<td></td>
<td>Specifies that the command should list URLs instead of names.</td>
</tr>
</tbody>
</table>

See Also

import_webdataconnector

delete_webdataconnector

Web Data Connectors in Tableau Server
manage_global_credentials

Manages credentials for delegated data access on Tableau Server. Use this command to specify the credentials for a proxy user that is used to access a data source that does not support single-sign on via Kerberos.

Examples

tabadmin manage_global_credentials --add --server <server> --user <username> --password <password>

Add credentials for a server named my-server.

tabadmin manage_global_credentials --add --server my-server --user jsmith --password p@ssword

<table>
<thead>
<tr>
<th>Option (short)</th>
<th>Option (long)</th>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>--add</td>
<td></td>
<td></td>
<td>Add credentials for the specified server.</td>
</tr>
<tr>
<td>--remove</td>
<td></td>
<td></td>
<td>Remove credentials</td>
</tr>
<tr>
<td>--show</td>
<td></td>
<td></td>
<td>Show current credentials</td>
</tr>
<tr>
<td>-s</td>
<td>--server</td>
<td>server</td>
<td>Server for which credentials are being managed</td>
</tr>
<tr>
<td>-u</td>
<td>--username</td>
<td>user</td>
<td>User name for connecting to a server</td>
</tr>
<tr>
<td>-p</td>
<td>--password</td>
<td>password</td>
<td>Password for connecting to a server</td>
</tr>
<tr>
<td>-o</td>
<td>--override</td>
<td></td>
<td>Override existing credentials</td>
</tr>
<tr>
<td>Option (short)</td>
<td>Option (long)</td>
<td>Argument</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>--------------</td>
<td>----------</td>
<td>-------------</td>
</tr>
<tr>
<td>ride</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

See Also

**Enabling Delegation for Cloudera Impala** in the Tableau Community.

### passwd

Resets the password for a Tableau Server account. After typing the command, you are prompted to enter a new password for the user.

You can only use this command if Tableau Server's user authentication is set to Local Authentication. When authentication is set to Active Directory, passwords are handled by Active Directory, not Tableau Server.

**Examples**

```
tabadmin passwd <username>
```

Reset the password for server user **jsmith**:

```
tabadmin passwd jsmith
```

See Also

Configure General Server Options
recommission

Reverts a decommissioned file store node in read-only mode to an active read/write state. Use spaces to separate multiple nodes.

**Examples**

tabadmin recommission <computer name(s) or IPv4 address(es)>

Recommission file store node by IP address:

tabadmin recommission 10.32.139.29

See Also

Distributed Environments

Maintain a Distributed Environment

**regenerate_internal_tokens**

Creates new security tokens that Tableau Server uses internally. These tokens include the passwords used by Tableau Server to access the repository, and the certificates used to validate internal SSL connections between Tableau Server components and the repository.

Running this command stops Tableau Server, so you will need to restart Tableau Server after you run the command.

**Example**

tabadmin regenerate_internal_tokens --passwords

`tabadmin regenerate_internal_tokens --certs`
<table>
<thead>
<tr>
<th>Option (short)</th>
<th>Option (long)</th>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>--certs</td>
<td>None</td>
<td>Regenerates key pair for internal SSL connections.</td>
<td></td>
</tr>
<tr>
<td>--passwords</td>
<td>None</td>
<td>Regenerates passwords for the Postgres database.</td>
<td></td>
</tr>
<tr>
<td>None</td>
<td>None</td>
<td>Regenerates key pair for internal SSL connections and passwords for Postgres database. <strong>Note:</strong> The key pair is regenerated only if internal SSL is configured.</td>
<td></td>
</tr>
<tr>
<td>--roll_key</td>
<td>None</td>
<td>Updates the secrets for encrypted passwords that are stored in the repository. See Manage Server Secrets.</td>
<td></td>
</tr>
<tr>
<td>--restart</td>
<td>None</td>
<td>Restart Tableau Server after regenerating tokens.</td>
<td></td>
</tr>
</tbody>
</table>

**See Also**

Regenerate a Password for the Tableau Server PostgreSQL Database (Repository)

**Security**

**register**

Register Tableau Server with the user information provided in the specified file. This is used for scripted, automated installations.

**Examples**

tabadmin register --file tabregister.json
<table>
<thead>
<tr>
<th>Option (short)</th>
<th>Option (long)</th>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>--file</td>
<td>JSON-format file</td>
<td></td>
<td>Register Tableau Server using the information in the specified file.</td>
</tr>
</tbody>
</table>

See Also

Automated Installation or Upgrade of Tableau Server

Registration Input File Options

reindex

Rebuilds the search index for Tableau Server. In rare instances, you may need to rebuild the index if searches on the server return incomplete or incorrect results, or if the Search & Browse process is down for an extended period. You can use this command if users cannot sign in to the server because no sites are listed after they enter their credentials.

**Note:** The recommended way to reindex Search is to run this command while Tableau Server is stopped. Reindexing while the server is running can result in content, including sites and projects, temporarily disappearing.

**Examples**

tabadmin reindex

Reindex the server

See Also

Rebuild the Search Index
reset

Resets the Tableau Server administrator account. This command will reset the server so that you will need to set up an administrator account. After resetting the administrator account you need to use a browser to connect to localhost and create a new administrator account.

Example

tabadmin reset

<table>
<thead>
<tr>
<th>Option (short)</th>
<th>Option (long)</th>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>--destroy-sessions</td>
<td>None</td>
<td>Destroys all existing sessions. All users will be forced to sign in again.</td>
<td></td>
</tr>
<tr>
<td>--silent</td>
<td>None</td>
<td>Suppresses normal verbose mode. This is useful if you are creating a chain of several automated steps.</td>
<td></td>
</tr>
</tbody>
</table>

See Also

Add an Administrator Account

reset_openid_sub

Clears the user identifier (sub value) that binds a user identity in Tableau Server to a specific OpenID Connect identity provider (IdP).

If Tableau Server is configured to use OpenID Connect for authentication, the first time a user signs in to Tableau Server using the IdP, Tableau stores the sub value sent by the IdP with the user information in Tableau Server. The sub provides a unique identity for that user
with the IdP. If you change IdPs for OpenID Connect, you must remove the sub value for the user. That way, when the user signs in using the new IdP, Tableau can store a new sub value.

<table>
<thead>
<tr>
<th>Option (short)</th>
<th>Option (long)</th>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>--username</td>
<td>username</td>
<td>Removes the sub value for the specified user.</td>
</tr>
<tr>
<td>--all</td>
<td>None</td>
<td>None</td>
<td>Removes the sub value for all users.</td>
</tr>
</tbody>
</table>

**Example**

`tabadmin reset_openid_sub --username Alice`

This command clears the sub value for the user named Alice.

`tabadmin reset_openid_sub --all`

This command clears the sub value for all users on the server.

**See Also**

OpenID Connect

Changing IdPs in Tableau Server for OpenID Connect

**restart**

Stops and starts all Tableau Server processes. The restart command also does a configuration so you do not need to do a `tabadmin config` if you are doing a restart (a config will not do any harm).

**Example**
tabadmin restart

<table>
<thead>
<tr>
<th>Option (short)</th>
<th>Option (long)</th>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>--skip_core_license_check</td>
<td></td>
<td></td>
<td>Skip the core license check when restarting Tableau Server. If used when licenses are not core-based, the option has no impact. (Added in version 10.5.10.)</td>
</tr>
</tbody>
</table>

restore

Restores a Tableau Server backup file (.tsbak) to a Tableau Server installation. When you restore a .tsbak file, the contents of the Tableau PostgreSQL database, extracts, and configuration files are overwritten with the content in the backup file. If the backup was made after web data connectors were imported to the server using the import_web-dataconnector command, the restore process restores the connectors as well. Using the --no-config option restores everything but the server’s configuration.

Examples

Restore a file named tabserv.tsbak located in C:\mybackups and then restart the server:

```
tabadmin restore C:\mybackups\tabserv.tsbak --restart
```

Restore a file named tabserv.tsbak located in the Tableau Server bin directory and then restart the server:

```
tabadmin restore tabserv.tsbak --restart
```

Restore a file named tabserv.tsbak located in C:\mybackups, retaining everything but the server’s configuration, but don’t restart the server:

```
tabadmin restore --no-config C:\mybackups\tabserv.tsbak
```
<table>
<thead>
<tr>
<th>Option (short)</th>
<th>Option (long)</th>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>--no-config</td>
<td>--no-config</td>
<td></td>
<td>Restore the Tableau Server backup file including the data but excluding the server's configuration.</td>
</tr>
<tr>
<td>--parallel-pg-restore</td>
<td>--parallel-pg-restore</td>
<td></td>
<td>Run the restore process for the PostgreSQL repository as a parallel job.</td>
</tr>
<tr>
<td>--password</td>
<td>--password</td>
<td>&lt;password&gt;</td>
<td>Restore the Tableau Server backup file using the Run As User password.</td>
</tr>
<tr>
<td>--password-file</td>
<td>--password-file</td>
<td>File</td>
<td>Restore the Tableau Server backup file, reading the password from the specified file.</td>
</tr>
<tr>
<td>--restart</td>
<td>--restart</td>
<td></td>
<td>Restart the service when the restore process has completed.</td>
</tr>
</tbody>
</table>

See Also

Restore from a Backup

Recover Extracts from a Backup

set

Allows you to change the value of Tableau Server configuration options.

In some cases, a set command can take multiple values. If so, the values are separated by commas and spaces.

After changing configuration values, you need to run a `tabadmin config` command to update configuration files on all Tableau Server nodes.
Examples

tabadmin set [option-name] [value]

tabadmin set --stdin [option-name] [value]

If the parameter you’re setting begins with a hyphen, enclose the parameter’s value in both double- and single-quotes.

```
tabadmin set [option-name] "'[-value]''
```

Set the backgrounder query limit to 2.5 hours (9000 seconds):

```
tabadmin set backgrounder.querylimit 9000
```

Disable sorting, based on duration of last run, of extract refresh jobs:

```
tabadmin set backgrounder.sort_jobs_by_run_time_history_observable_hours "'-1'"
```

Set the names of two web servers that request trusted tickets from Tableau Server.

```
tabadmin set wgserver.trusted_hosts myserver1, myserver2
```

<table>
<thead>
<tr>
<th>Option (short)</th>
<th>Option (long)</th>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-d</td>
<td>--default</td>
<td></td>
<td>Reset the parameter to its default value.</td>
</tr>
<tr>
<td>-s</td>
<td>--stdin</td>
<td></td>
<td>Read a parameter value from the standard input stream (STDIN) such as the console. You are prompted for the value.</td>
</tr>
</tbody>
</table>

See Also

```
tabadmin configuration options
```

- 1148 -
sitestate

Activates (unlocks) or suspends a site. You can use this command to activate a site that was locked because of a site import failure. When a site is suspended, the only Tableau Server user who can access it is the system administrator.

**Note:** To specify the default site, use "" for the site ID.

**Examples**

```
tabadmin sitestate <site ID> --status <active|suspended>
```

Activate a site whose site ID is **wsales**:

```
tabadmin sitestate wsales --status active
```

Activate the Default site. The site ID for the Default site is "" (double quotes, no space).

```
tabadmin sitestate "" --status active
```

<table>
<thead>
<tr>
<th>Option (short)</th>
<th>Option (long)</th>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>--status</td>
<td></td>
<td>active</td>
<td>Specifies whether to activate or suspend the specified site.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>or</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>suspended</td>
<td></td>
</tr>
</tbody>
</table>
start

Starts all Tableau Server processes. The start command also does a configuration so you do not need to do a `tabadmin config` if you are doing a start (a config will not do any harm).

To use `tabadmin start`:

1. Open a command prompt as an administrator:

```
Open
Run as administrator
7-Zip
Select Left File for Compare
```

2. Type the following:

```bash
cd "C:\Program Files\Tableau\Tableau Server\10.5\bin"
```

3. Type the following to start the server:

```bash
tabadmin start
```

Examples

```bash
tabadmin start

tabadmin start --wait 1200
```
<table>
<thead>
<tr>
<th>Option (short)</th>
<th>Option (long)</th>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>--skip_core_license_check</td>
<td></td>
<td></td>
<td>Skip the core license check when starting Tableau Server. If used when licenses are not core-based, the option has no impact. (Added in version 10.5.10.)</td>
</tr>
<tr>
<td>--wait</td>
<td>number of seconds</td>
<td></td>
<td>Number of seconds after starting after which Tableau Server is ready to accept client requests. The default is 600 seconds.</td>
</tr>
</tbody>
</table>

**status**

Tells you whether or not Tableau Server is running and, if you use the `--verbose` option, gives you details on individual server process status, including whether a process is running and its process ID. The `tabadmin status` command obtains its information by connecting to the Windows Service `tabsvc.exe`, which in turn queries the `tabspawn` executables for each process. Because of this, it can sometimes display different information for the server processes than the status table on the Maintenance page, which queries the processes directly.

**Examples**

```
tabadmin status

tabadmin status --verbose
```

<table>
<thead>
<tr>
<th>Option (short)</th>
<th>Option (long)</th>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-v</td>
<td>--verbose</td>
<td></td>
<td>Returns a list of all the Tableau Server processes, their process IDs, and their status.</td>
</tr>
</tbody>
</table>
See Also

Server Settings (General)

Tableau Server Processes

**stop**

Stops all Tableau Server processes.

**Note:** Using the Windows Task Manager to stop Tableau Server can cause problems with your Tableau Server installation. Instead, you should always use the `tabadmin stop` command to stop Tableau Server.

**To use** `tabadmin stop`:

1. Open a command prompt as an administrator:

   ![Command Prompt](image)

2. Type the following:
cd "C:\Program Files\Tableau\Tableau Server\10.5\bin"

3. Type the following to stop the server:

    tabadmin stop

**validate**

Confirms whether your Tableau Server environment meets the minimum requirements for running Tableau Server.

**Example**

tabadmin validate

<table>
<thead>
<tr>
<th>Option (short)</th>
<th>Option (long)</th>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>--skiptemp-pIPv6</td>
<td>--skiptem-pIPv6</td>
<td></td>
<td>Skip validating that temporary IPv6 addresses are disabled.</td>
</tr>
</tbody>
</table>

**validate_resources**

 Validates workbooks and data sources for a site.

**Example**

tabadmin validate_resources <site ID>

To specify the default site, use empty double quotes ("" ) for the site ID. For example:

tabadmin validate_resources ""
<table>
<thead>
<tr>
<th>Option (short)</th>
<th>Option (long)</th>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>--repair</td>
<td></td>
<td></td>
<td>Repair invalid resources.</td>
</tr>
</tbody>
</table>

verify_database

Verifies that a backup of the PostgreSQL database that serves as the Tableau Server repository will restore successfully.

**Note:** The `verify_database` command is available beginning with Tableau Server version 9.3.

If you specify a backup file (.tsbak) as an option, the command restores the file to a temporary database in order to verify the backup. If you do not specify a backup file, a temporary backup of the running database is created and then restored to a temporary database. If verification fails, errors are displayed on the command line and are also logged in the `tabadmin.log` log file. Until the errors are addressed, You cannot restore a .tsback file that fails verification. If verification of the database fails, contact Tableau Support for assistance.

**Note:** A running PostgreSQL database can have errors that don't impact use but would cause a failure when you tried to restore a backup. This means that you may be able to continue to use a running database, but you cannot back it up and restore the backup. As a best practice, verify your database before taking a backup (prior to an upgrade, for example).

**Example**

Verify a backup file in the `C:\backups\tableau` folder named `tabserv.tsbak`: 
**tabadmin verify_database --file** C:|--backups\tableau\tabserv.tsbak

or

**tabadmin verify_database --file** "C:\Program Files\Tableau\Tableau Server\10.5\bin\tabserv.tsbak"

<table>
<thead>
<tr>
<th>Option (short)</th>
<th>Option (long)</th>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-f</td>
<td>--file</td>
<td>FILE</td>
<td>Backup file to verify. If no file is specified, the command verifies the running PostgreSQL database by making a temporary backup of it.</td>
</tr>
<tr>
<td>-t</td>
<td>--temp</td>
<td>PATH</td>
<td>Location of the temporary folder to use while doing verification. The default is the Tableau temp folder. Temporary files are removed after verification completes.</td>
</tr>
</tbody>
</table>

If the path includes spaces, use quotes around the full path: "C:\Program Files\Tableau\Tableau Server\10.5\bin\tabserv.tsbak"

**See Also**

backup

Verify the Tableau Postgres Database

**warmup**

**Note:** This command has been deprecated and has no effect. Initial load times were improved, making this command unnecessary.
Causes every VizQL server process to load the vizql DLL file, resulting in faster load times when server users first load views. Administrators can run this command, or script it to be run, after a Tableau Server restart.

**Example**

```bash
tabadmin warmup
```

**whitelist_webdataconnector**

Adds a web data connector to the safe list (whitelist) for an installation of Tableau Server. Users who create workbooks can then reference the web data connector as a data source. Tableau Server only uses the safe list if the `tabadmin web-dataconnector.whitelist.mode` property is set to `fixed` or `mixed`.

The safe list includes the URLs of hosted web data connectors that you have vetted and that you want to allow Tableau Server users to connect to. For more information, see Web Data Connectors in Tableau Server.

**Important:** Before you add a web data connector to the safe list, check the functionality of the connector. For more information, see Testing and Vetting Web Data Connectors.

The URLs for connectors in the safe list are case sensitive. When a connector is on the safe list, data sources and workbooks that are associated with the connector can be refreshed on Tableau Server. After you make changes to the safe list, you must restart Tableau Server for the changes to take effect.

For each connector that you add to the safe list, you must also create a secondary safe list specific to that connector. This secondary safe list determines which domains the connector can send requests to and receive requests from. This helps ensure that connectors do not send information to untrusted domains.
<table>
<thead>
<tr>
<th>Option (short)</th>
<th>Option (long)</th>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-a</td>
<td>--add</td>
<td>Connector URL</td>
<td>Add the URL of a hosted web data connector to the safe list.</td>
</tr>
<tr>
<td>-d</td>
<td>--delete</td>
<td>Connector URL</td>
<td>Remove the URL of a web data connector from the safe list.</td>
</tr>
<tr>
<td>-l</td>
<td>--list</td>
<td></td>
<td>List all of the connector URLs on the safe list.</td>
</tr>
<tr>
<td>-r</td>
<td>--reset</td>
<td></td>
<td>Clear the safe list.</td>
</tr>
<tr>
<td>-s</td>
<td>--add_secondary_whitelist</td>
<td>Connector URL and comma separated list of URLs the connector can make requests to</td>
<td>Add a list of domains that a particular connector can make requests to. The first argument is a connector that has already been added to the safe list. The second argument is a comma separated list of domains or resources. You can include regular expressions in the list of domains. If you do not specify a secondary safe list for a connector, then that connector cannot connect to any domains.</td>
</tr>
<tr>
<td>-p</td>
<td>--print_secondary_whitelist</td>
<td>Connector URL</td>
<td>Print the secondary safe list for a given connector URL.</td>
</tr>
<tr>
<td>-w</td>
<td>--reset_secondary_whitelist</td>
<td>Connector URL</td>
<td>Clear the secondary safe list for a given connector URL.</td>
</tr>
</tbody>
</table>

**Note:** If you use Windows PowerShell to run these commands, you might need to include quotes around the argument. For example, when you add a secondary safe list with regular expressions, you need to include quotes.
Examples

tabadmin whitelist_webdataconnector -a https://example.com/myconnector.html

tabadmin whitelist_webdataconnector -a http://example.com:8080/myconnector.html

tabadmin whitelist_webdataconnector -d https://example.com/myconnector.html


tabadmin whitelist_webdataconnector -p https://example.com/myconnector.html

tabadmin whitelist_webdataconnector -w https://example.com/myconnector.html

See Also

Web Data Connectors in Tableau Server

Tableau webdataconnector page on GitHub

ziplogs

Creates an archive (.zip) containing Tableau Server log files, without removing the log files themselves. If you are running a Tableau Server cluster, log files from worker servers are included in the archive that’s created. You can specify the file name and location. If you don’t provide a file name, the archive is called logs.zip. If you don’t include a path, the file is saved in the \bin directory. Saving an archive file to a UNC path is not supported.

Examples

Create an archive in the Tableau Server bin directory named logs.zip:

tabadmin ziplogs
Create an archive in the Tableau Server bin directory named \texttt{mylogs.zip}:

\texttt{tabadmin ziplogs mylogs.zip}

Create an archive in the Tableau Server bin directory named \texttt{mylogs.zip} that includes logs dated January 31, 2014 up to the present, excluding earlier logs:

\texttt{tabadmin ziplogs -d 01/31/2014 mylogs.zip}

<table>
<thead>
<tr>
<th>Option (short)</th>
<th>Option (long)</th>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>-n</td>
<td>--with-net-stat-info</td>
<td></td>
<td>Include information about the server environment in the .zip file.</td>
</tr>
<tr>
<td>-p</td>
<td>--with-postgresql-data</td>
<td></td>
<td>Include data from Tableau Server's PostgreSQL database. If Tableau Server is stopped, make a copy of the psql\data folder. If Tableau Server is running, get the data as binary dump files.</td>
</tr>
<tr>
<td>-l</td>
<td>--with-latest-dump</td>
<td></td>
<td>Limit the included log files to only the most recent ones to help reduce file size. By default, the 10 most recent log files are included.</td>
</tr>
<tr>
<td>-f</td>
<td>--force</td>
<td></td>
<td>Overwrites the existing log file of the same name.</td>
</tr>
<tr>
<td>-d</td>
<td>--min-imumdate</td>
<td>[mm/dd/yyyy]</td>
<td>Log files with this date, up to the present, are included in the .zip file. Logs dated earlier are excluded from the file. If not specified, up to seven days worth of data is included.</td>
</tr>
<tr>
<td>-a</td>
<td>--all</td>
<td></td>
<td>Include all log files in the .zip file. Data from Tableau Server's PostgreSQL database is still excluded.</td>
</tr>
</tbody>
</table>
See Also

Work with Log Files

Archive Logs on Command Line (tabadmin)

tabadmin configuration options

Use the table below to learn more about Tableau Server options you can configure using the set command and query using the get. See Tableau Server Ports for a complete list of ports.

After using `tabadmin set` to change configuration options, you need to run a `tabadmin config` command to update configuration files on all Tableau Server nodes.

<table>
<thead>
<tr>
<th>Option</th>
<th>Default Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>api.server.enabled</td>
<td>true</td>
<td>Allows access to the Tableau Server REST API. By default, this functionality is enabled.</td>
</tr>
<tr>
<td>auditing.enabled</td>
<td>true</td>
<td>Allows access to the PostgreSQL (Tableau Server's own database) historical auditing tables. See Collect Data with the Tableau Server Repository for details.</td>
</tr>
<tr>
<td>back-ground-er.externalquerycachewarmup.enabled</td>
<td>true</td>
<td>Controls the caching of workbook query results after scheduled extract refresh tasks. See Configure Workbook Performance after a Scheduled Refresh.</td>
</tr>
<tr>
<td>back-ground-er.externalquerycachewarmup.view_</td>
<td>2.0</td>
<td>The threshold for caching workbook query results after scheduled extract refresh tasks. The</td>
</tr>
<tr>
<td>Option</td>
<td>Default Value</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------------------------</td>
<td>---------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>threshold</td>
<td></td>
<td>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. See Configure Workbook Performance after a Scheduled Refresh.</td>
</tr>
<tr>
<td>backgrounder.extra_timeout_in_seconds</td>
<td>1800</td>
<td>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.</td>
</tr>
</tbody>
</table>
| backgrounder.failure_threshold_for_run_prevention | 5             | The number of consecutive failures of a subscription or extract 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.  

**Note:** To reenable a suspended job, click **Try again** from the alert menu, or republish the data.
<table>
<thead>
<tr>
<th>Option</th>
<th>Default Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>source or a workbook using the data source, or change the connection properties of the data source.</td>
</tr>
<tr>
<td>backgrounder.querylimit</td>
<td>7200</td>
<td>Longest allowable time, in seconds, for completing a single extract refresh task or subscription task. 7200 seconds = 2 hours.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Note:</strong> If a background task reaches this time limit, it may continue to run for an additional several minutes while being canceled.</td>
</tr>
<tr>
<td>backgrounder.reset_schedules_on_startup</td>
<td>true</td>
<td>Controls when to run background tasks that were scheduled to run at a time when the server was stopped. When set to true (the default), tasks are run at their next scheduled time. When set to false, all tasks that were scheduled to run when the server was stopped are run, simultaneously, at server startup, including times when the Tableau Server backup file (.tsbak) is restored.</td>
</tr>
<tr>
<td>backgrounder.send_email_on_refresh_failure</td>
<td>true</td>
<td>Controls whether extract refresh</td>
</tr>
<tr>
<td>Option</td>
<td>Default Value</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------------------------</td>
<td>---------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>alerts</td>
<td></td>
<td>alerts are enabled for all sites on the server. By default alerts are enabled. To disable extract refresh alerts for all sites on a server, set this to <code>false</code>. 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.</td>
</tr>
<tr>
<td><code>backgrounder.sort_jobs_by_run_time_history_observable_hours</code></td>
<td>-1</td>
<td>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 &quot;last run,&quot; executing the fastest full extract refresh jobs first. The &quot;last run&quot; duration of a particular job is determined from a random sample of a single instance of the full extract refresh job in last <code>&lt;n&gt;</code> hours. Full extract jobs are then prioritized to run in order from shortest to longest based on their &quot;last&quot; run duration. By default this is sorting is</td>
</tr>
<tr>
<td>Option</td>
<td>Default Value</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------------------------------------------------------</td>
<td>---------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>disabled (-1). If enabling this, the suggested value is 36 (hours).</td>
<td>60000</td>
<td>Controls the time window that identifies backgrounder jobs which are determined to have the same scheduled start time. 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 &quot;same time&quot;. 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.</td>
</tr>
<tr>
<td>backgrounder.sort_jobs_by_type_schedule_boundary_heuristics_milliSeconds</td>
<td>true</td>
<td>Controls whether backgrounder will cache images that are generated for subscriptions. Cached images do not have to be regenerated each time so caching</td>
</tr>
<tr>
<td>Option</td>
<td>Default Value</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------------------</td>
<td>--------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>'improves subscription performance. By default image caching is enabled. To disable image caching for all sites on a server, set this to false.'</td>
<td></td>
<td></td>
</tr>
<tr>
<td>backgrounder.timeout_tasks</td>
<td>refresh_extracts, increment_extracts, subscription_notify, single_subscription_notify</td>
<td>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.</td>
</tr>
<tr>
<td>cluster-controller.pgsql.failover</td>
<td>true</td>
<td>In a high availability environment, controls whether failover of the PostGRES repository occurs automatically (the default). When set to false, failover to the passive repository only occurs when you to run the failoverrepository command.</td>
</tr>
<tr>
<td>clustercontroller.zk_session_timeout_ms</td>
<td>300000</td>
<td>The length of time, in milliseconds, that Cluster Controller will wait for the Coordination Service (ZooKeeper), before determining that failover is required.</td>
</tr>
<tr>
<td>dataAlerts.checkIntervalInMinutes</td>
<td>60</td>
<td>The frequency, in minutes, at which Tableau Server checks to</td>
</tr>
<tr>
<td>Option</td>
<td>Default Value</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------------------------</td>
<td>---------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td></td>
<td></td>
<td>determine if data-alert conditions are true. (The server also checks whenever extracts related to data alerts are refreshed.)</td>
</tr>
<tr>
<td>dataAlerts.retryFailedAlerts-AfterCheckInterval</td>
<td>true</td>
<td>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.</td>
</tr>
<tr>
<td>dataengine.port</td>
<td>27042</td>
<td>Port that the data engine runs on.</td>
</tr>
<tr>
<td>dataserver.port</td>
<td>9700</td>
<td>Port that the data server runs on.</td>
</tr>
<tr>
<td>DataServer-RefreshMetadataPer-Session</td>
<td>false</td>
<td>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 per-</td>
</tr>
<tr>
<td>Option</td>
<td>Default Value</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------------------------</td>
<td>---------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>PerformanceReasons</td>
<td></td>
<td>formance 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, &quot;An error occurred while communicating with the data source: Invalid column name. Statement could not be prepared.&quot;) set this to true. When set to true, Tableau Server makes additional queries to update the schema.</td>
</tr>
<tr>
<td>features.Aler tOnThresholdCondition</td>
<td>true</td>
<td>Controls whether data-driven alerts are enabled for users on the server.</td>
</tr>
<tr>
<td>features.DesktopReporting</td>
<td>false</td>
<td>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.</td>
</tr>
<tr>
<td>gateway.http.cachecontrol.updated</td>
<td>false</td>
<td>The Cache-Control HTTP header specifies whether the client browser should cache con-</td>
</tr>
<tr>
<td>Option</td>
<td>Default Value</td>
<td>Description</td>
</tr>
<tr>
<td>----------------------</td>
<td>---------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td></td>
<td></td>
<td>tent sent from Tableau Server. To disable caching of Tableau Server data on the client, set this option to <code>true</code>.</td>
</tr>
<tr>
<td>gateway.http.hsts</td>
<td><code>false</code></td>
<td>The HTTP Strict Transport Security (HSTS) header forces browsers to use HTTPS on the domain where it is enabled.</td>
</tr>
<tr>
<td>gateway.http.hsts_options</td>
<td>&quot;max-age=31536000&quot;</td>
<td>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.</td>
</tr>
<tr>
<td>gateway.http.request_size_limit</td>
<td>16380</td>
<td>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 <code>gateway.http.request_size_limit</code> may result in authentication errors. Single sign-on solutions that integrate with Active Directory (SAML and Ker-</td>
</tr>
<tr>
<td>Option</td>
<td>Default Value</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>---------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td></td>
<td></td>
<td>beros) often require large authentication tokens in HTTP headers. Be sure to test HTTP authentication scenarios before deploying into production. We recommend setting <code>tomcat.http.maxrequestsize</code> option to the same value that you set for this option.</td>
</tr>
<tr>
<td>gateway.http.x_content_type_nosniff</td>
<td>true</td>
<td>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 &quot;sniffing.&quot; 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.</td>
</tr>
<tr>
<td>gateway.http.x_xss_pro</td>
<td>true</td>
<td>The HTTP X-XSS-Protection</td>
</tr>
<tr>
<td>Option</td>
<td>Default Value</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------</td>
<td>---------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>xss-protection</td>
<td></td>
<td>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.</td>
</tr>
<tr>
<td>gateway.public.host</td>
<td>Name of the machine</td>
<td>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 <code>tableau-example.com</code>, the name for <code>gateway.public.host</code> is <code>tableau-example.com</code>.</td>
</tr>
<tr>
<td>gateway.public.port</td>
<td>80 (443 if SSL)</td>
<td>Applies to proxy server environments only. The external port the proxy server listens on.</td>
</tr>
<tr>
<td>gateway.slow_post_protection.enabled</td>
<td>false</td>
<td>Enabling this can provide some help in protecting against slow POST (Denial-of-Service) attacks by timing out POST.</td>
</tr>
<tr>
<td>Option</td>
<td>Default Value</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>----------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>requests</td>
<td></td>
<td>requests that transfer data at extremely slow rates. <strong>Note:</strong> This will not eliminate the threat of such attacks, and could have the unintended impact of terminating slow connections.</td>
</tr>
<tr>
<td>gateway.timeout</td>
<td>1800</td>
<td>Longest amount of time, in seconds, that the gateway will wait for certain events before failing a request (1800 seconds = 30 minutes).</td>
</tr>
<tr>
<td>gateway.trusted</td>
<td>IP address of proxy server machine</td>
<td>Applies to proxy server environments only. The IP address(es) or host name(s) of the proxy server.</td>
</tr>
<tr>
<td>gateway.trusted_hosts</td>
<td>Alternate names of proxy server</td>
<td>Applies to proxy server environments only. Any alternate host name(s) for the proxy server.</td>
</tr>
<tr>
<td>install.firewall.allowedprograms.manage</td>
<td>true</td>
<td>Controls whether Tableau Server can add firewall rules. When set to <strong>true</strong> (the default), Tableau Server will add new firewall rules to allow its processes to make connections through Windows Firewall. Change this to <strong>false</strong> if you want to manage all firewall rules yourself and do not want Tableau Server to add new rules.</td>
</tr>
<tr>
<td>Option</td>
<td>Default Value</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------------------------</td>
<td>---------------</td>
<td>------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>java.heap.size</td>
<td>128m</td>
<td>Size of heap for Tomcat (repository and solr). This generally does not need to change except on advice from Tableau.</td>
</tr>
<tr>
<td>monitoring.dataengine.connection_timeout</td>
<td>30000</td>
<td>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).</td>
</tr>
<tr>
<td>native_api.-connection.limit.&lt;connection class&gt;</td>
<td></td>
<td>Set parallel query limit for the specified data source (connection class). This overrides the global limit for the data source.</td>
</tr>
<tr>
<td>native_api.-connection.globallimit</td>
<td>16</td>
<td>&gt;Global limit for parallel queries. When not specified, the default is 16 (except for Amazon Redshift, which has a default of 8). When specified, the limit applies to all data source types.</td>
</tr>
<tr>
<td>native_api.ProtocolTransitionLegacyFormat</td>
<td>false</td>
<td>Use the legacy name format for constrained delegation.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The name format was changed in version 10.1 to allow cross-domain protocol transition (S4U). If this causes problems with existing configurations and</td>
</tr>
<tr>
<td>Option</td>
<td>Default Value</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------------------------</td>
<td>---------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>you don’t need cross-domain protocol</td>
<td></td>
<td>transition, configure Tableau Server to use the old behavior by setting this to <code>true</code>.</td>
</tr>
<tr>
<td>native_api.recycling_interval_minutes</td>
<td>1440</td>
<td>Sets the number of minutes after which a process can be restarted, or &quot;recycled&quot;, if thread recycling is enabled.</td>
</tr>
<tr>
<td>native_api.recycling_thread_enabled</td>
<td>false</td>
<td>Enables thread recycling.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Thread recycling restarts processes on a fixed schedule regardless of their resource usage, and separate from any restarts caused by exceeding</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CPU or memory limits. Thread recycling schedules are set using <code>native_api.recycling_time_start</code> and <code>native_api.recycling_time_end</code>.</td>
</tr>
<tr>
<td>native_api.recycling_time_start</td>
<td>1380</td>
<td>Sets the scheduled start time for thread recycling, if enabled.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Specify the time using a number of minutes since the start of the day, with no quotes required.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The default value of 1380 is the equivalent of 11 PM (23:00).</td>
</tr>
<tr>
<td>Option</td>
<td>Default Value</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------------------</td>
<td>---------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>native_api.recycling_time_end</td>
<td>240</td>
<td>Sets the scheduled end time for thread recycling, if enabled. Specify the time using a number of minutes since the start of the day, without quotes. The default value of 240 is the equivalent of 4 AM (04:00).</td>
</tr>
<tr>
<td>features.PasswordReset</td>
<td>false</td>
<td>Applies only to servers that use local authentication. Set to <code>true</code> to let users reset their passwords with a &quot;Forgot password&quot; option on the sign-in page.</td>
</tr>
<tr>
<td>pgsql.port</td>
<td>8060</td>
<td>Port that PostgreSQL listens on.</td>
</tr>
<tr>
<td>pgsql.verify_restore.port</td>
<td>8061</td>
<td>Port used to verify the integrity of the PostgreSQL database. See Verify the Tableau Postgres Database for more information.</td>
</tr>
<tr>
<td>recommendations.enabled</td>
<td>true</td>
<td>Suggests server content, such as data sources and tables, to Tableau Desktop users. Content suggestions are based on popularity of the content or on content frequently used by other users who are similar to the current user.</td>
</tr>
<tr>
<td>refresh_token.absolute_expiry_in_seconds</td>
<td>31536000</td>
<td>Specifies the number of seconds for absolute expiry of OAuth refresh and access tokens. The</td>
</tr>
<tr>
<td>Option</td>
<td>Default Value</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------------------------</td>
<td>---------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>refresh_token_idle_expiry_in_seconds</td>
<td>1209600</td>
<td>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).</td>
</tr>
<tr>
<td>refresh_token_max_count_per_user</td>
<td>24</td>
<td>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.</td>
</tr>
<tr>
<td>rsync.timeout</td>
<td>600</td>
<td>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.</td>
</tr>
<tr>
<td>schedules.display_schedule_description_as_name</td>
<td>false</td>
<td>Controls whether a schedule</td>
</tr>
<tr>
<td>Option</td>
<td>Default Value</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------------------------</td>
<td>---------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>name displays when creating a subscription</td>
<td></td>
<td>name displays when creating a subscription or extract refresh (the default), or the &quot;schedule frequency description&quot; 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 &quot;schedule frequency description&quot; is also displayed after the schedule name on the schedule list page.</td>
</tr>
<tr>
<td>schedules.display_schedules_in_client_timezone</td>
<td>true</td>
<td>Shows the &quot;schedule frequency description&quot; in the timezone of the user when true (uses the client browser timezone to calculate the &quot;schedule frequency description&quot;).</td>
</tr>
<tr>
<td>searchserver.index.bulk_query_user_groups</td>
<td>true</td>
<td>When used with viz-portal.csv_user_mgmt.bulk_index_users set to true, this determines whether the indexer queries groups in bulk or one at a time. The default value of true</td>
</tr>
<tr>
<td>Option</td>
<td>Default Value</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------------------</td>
<td>---------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>means the groups will be queried in bulk.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>service.init.state</td>
<td>start</td>
<td>Determines whether or not Tableau Server will automatically start when operating system of the computer Tableau Server is running on is restarted. Valid options are start and pause. Set this to pause if Tableau Server should not start on a restart of the computer.</td>
</tr>
<tr>
<td>service.jmx_enabled</td>
<td>false</td>
<td>Setting to true enables JMX ports for optional monitoring and troubleshooting. See Enable the JMX Ports for details.</td>
</tr>
<tr>
<td>service.max_procs</td>
<td># of processes</td>
<td>Maximum number of server processes.</td>
</tr>
<tr>
<td>service.port_remappping.enabled</td>
<td>true</td>
<td>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. See Tableau Server Ports for more information.</td>
</tr>
<tr>
<td>Option</td>
<td>Default Value</td>
<td>Description</td>
</tr>
<tr>
<td>-------------</td>
<td>---------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| session.ipsticky | 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
<table>
<thead>
<tr>
<th>Option</th>
<th>Default Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>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.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>sheet_image.enabled</td>
<td>true</td>
<td>Controls whether you can get images for views with the REST API. For more information, see the REST API Reference in the REST API help.</td>
</tr>
<tr>
<td>solr.rebuild_index_timeout</td>
<td>3600</td>
<td>When Tableau Server is upgraded or when a .tsbak file is restored, the background task rebuilds the search index. This setting controls the timeout setting for that task (3600 seconds = 60 minutes).</td>
</tr>
<tr>
<td>ssl.ciphersuite</td>
<td>HIGH:MEDIUM:!a-NULL:!MD5:!RC4</td>
<td>The Triple-DES cipher suite is</td>
</tr>
<tr>
<td>Option</td>
<td>Default Value</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------------------</td>
<td>---------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td></td>
<td></td>
<td>no longer considered adequate to encrypt sessions on the internet. Specifically, running Triple-DES ciphers leaves the Tableau Server vulnerable to information disclosure and denial of service attacks. You can learn more at the National Vulnerability Database webpage for <strong>CVE-2016-2183</strong>.</td>
</tr>
<tr>
<td>ssl.client_certificate_login.mapping_strategy</td>
<td>UPN</td>
<td>Specifies the method to be used for retrieving the user name from the certificate. The default depends on how Tableau Server is configured for user authentication:</td>
</tr>
<tr>
<td>Option</td>
<td>Default Value</td>
<td>Description</td>
</tr>
<tr>
<td>----------------</td>
<td>---------------------------------------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td></td>
<td></td>
<td>* When Tableau Server authentication is configured for Local Authentication, the default is <strong>UPN</strong> (User Principal Name).</td>
</tr>
<tr>
<td></td>
<td></td>
<td>* When Tableau Server authentication is configured for Active Directory (AD), the default is <strong>LDAP</strong> (Lightweight Directory Access Protocol).</td>
</tr>
<tr>
<td>CN</td>
<td></td>
<td><strong>CN</strong> (Common Name) is an option the administrator can set for either authentication type.</td>
</tr>
<tr>
<td>ssl.protocols</td>
<td><strong>all -SSLv2 -SSLv3</strong></td>
<td>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. Specially, 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,</td>
</tr>
<tr>
<td>Option</td>
<td>Default Value</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------</td>
<td>---------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>ssl.revocation.file</td>
<td></td>
<td>Specifies the file path for an SSL CA Certificate Revocation List (CRL) file. Example: <code>tabadmin set ssl.revocation.file &quot;c:\Program Files\Tableau\Tableau Server\SSL\ca-bundle-client.crl&quot;</code></td>
</tr>
<tr>
<td>subscriptions.enabled</td>
<td>false</td>
<td>Controls whether subscriptions are configurable system-wide. See Set Up a Server for Subscriptions.</td>
</tr>
<tr>
<td>Option</td>
<td>Default Value</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------</td>
<td>---------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>subscriptions.timeout</td>
<td>1800</td>
<td>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.</td>
</tr>
</tbody>
</table>
| tomcat.http.maxrequestsize | 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 |
<table>
<thead>
<tr>
<th>Option</th>
<th>Default Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>scenarios before deploying into production.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>We recommend setting</strong> <code>gateway.http.request_size_limit</code> <strong>option to the same value that you set for this option.</strong></td>
</tr>
<tr>
<td>tomcat.https.port</td>
<td>8443</td>
<td>SSL port for Tomcat (unused).</td>
</tr>
<tr>
<td>tomcat.server.port</td>
<td>8085</td>
<td>Port that tomcat listens on for shutdown messages.</td>
</tr>
<tr>
<td>vizportal.adsync.update_system_user</td>
<td>false</td>
<td>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 <code>vizportal.adsync.update_system_user</code> to <code>true</code>, and then restart the server.</td>
</tr>
<tr>
<td>vizportal.csv_user_mgmt_index_site_users</td>
<td>true</td>
<td>When you import or delete users through a CSV file and appropriate <code>tabcmd</code> command, this</td>
</tr>
<tr>
<td>Option</td>
<td>Default Value</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>---------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td></td>
<td></td>
<td>specifies how user indexing is accomplished. The default setting of true means indexing is done as each user is added or deleted. To improve performance, you can do either of the following:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Leave this set to true, and set the <code>vizportal.csv_user_mgmt.bulk_index_users</code> option to true. This is recommended because it does not need to index the whole site, and it can be used along with <code>search-server.index.bulk_query_user_groups</code>.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- Set this option to false. This indexes the whole site after the entire CSV file has been processed (including existing users that were not changed during the CSV process).</td>
</tr>
<tr>
<td>vizportal.csv_user_</td>
<td>false</td>
<td>Determines whether users</td>
</tr>
<tr>
<td>Option</td>
<td>Default Value</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------------</td>
<td>---------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>mgmt.bulk_index_users</td>
<td></td>
<td>imported or removed through a CSV file and appropriate <code>tab-cmd</code> command will be indexed individually or as a group. The default setting of <code>false</code> means that users are indexed one-by-one as they are added to the database. To improve performance when you are working with large sets of users, set this to <code>true</code>, to index users after the CSV file is processed. This option enables you also to use <code>search-server.index.bulk_query_user_groups</code> set to <code>true</code> for best performance results.</td>
</tr>
<tr>
<td>vizportal.log.level</td>
<td>info</td>
<td>The logging level for vizportal Java components. Logs are written to <code>ProgramData\Tableau\Tableau Server\data\t-absvc\logs\vizportal\*.log</code>.</td>
</tr>
<tr>
<td>Option</td>
<td>Default Value</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------------------</td>
<td>---------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>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. See Change Logging Levels for more information.</strong></td>
</tr>
<tr>
<td>vizportal.openid.client_authentication</td>
<td></td>
<td>Specifies custom client authentication method for OpenID Connect. To configure Tableau Server to use the Salesforce IdP, set this value to client_secret_post.</td>
</tr>
<tr>
<td>vizportal.openid.enabled</td>
<td>false</td>
<td>In Tableau Server 10.3, set to true to enable OpenID Connect SSO authentication.</td>
</tr>
<tr>
<td>vizportal.openid.id_claim</td>
<td>sub</td>
<td>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. For more information, see Requirements for Using OpenID Connect.</td>
</tr>
<tr>
<td>vizportal.openid.ignore_jwk</td>
<td>false</td>
<td>Set this to true if your IdP does</td>
</tr>
<tr>
<td>Option</td>
<td>Default Value</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------------------</td>
<td>---------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td></td>
<td></td>
<td>not support JWK validation. In this case, we recommend authenticating communication with your IdP using mutual TLS or another network layer secure protocol.</td>
</tr>
<tr>
<td>vizportal.openid.ignore_domain</td>
<td>false</td>
<td>Set this to true if the following are true:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• You are using email addresses as usernames in Tableau Server</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• You have provisioned users in the IdP with multiple domain names</td>
</tr>
<tr>
<td></td>
<td></td>
<td>• You want to ignore the domain name portion of the email claim from the IdP</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Before you proceed, review the user names that will be used as a result of setting vizportal.openid.ignore_domain 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.</td>
</tr>
<tr>
<td>Option</td>
<td>Default Value</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>---------------</td>
<td>-------------</td>
</tr>
<tr>
<td>vizportal.openid.static_file</td>
<td>file path</td>
<td>Specifies the local path to the static OIDC discovery JSON document. See Configure Tableau Server for OpenID Connect.</td>
</tr>
<tr>
<td>vizportal.openid.username_claim</td>
<td>email</td>
<td>Change this value to the IdP claim that your organization will use to match usernames as stored in Tableau Server. For more information, see Requirements for Using OpenID Connect.</td>
</tr>
<tr>
<td>vizportal.rest_api.-cors.allow_origin</td>
<td></td>
<td>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 (,).</td>
</tr>
</tbody>
</table>

```
```

If vizportal.rest_api.-cors.enabled is false, the origins listed by this option are ignored. For more information,
<table>
<thead>
<tr>
<th>Option</th>
<th>Default Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>see Enabling CORS on Tableau Server.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Note:</strong> You could also use an asterisk (<em>) 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 could present a security risk. Do not use an asterisk (</em>) unless you fully understand the implications and risks for your site.</td>
</tr>
<tr>
<td>vizportal.rest_api.-cors.enabled</td>
<td>false</td>
<td>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.</td>
</tr>
<tr>
<td>Option</td>
<td>Default Value</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------------------------</td>
<td>---------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td></td>
<td></td>
<td>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.</td>
</tr>
<tr>
<td>vizportal.rest_api.view_image.max_age</td>
<td>720</td>
<td>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.</td>
</tr>
<tr>
<td>vizqlserver.allow_insecure_scripts</td>
<td>false</td>
<td>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. You should set this value to true only if you want to use workbooks that contain SQL or R expressions that have been detected as potentially unsafe,</td>
</tr>
<tr>
<td>Option</td>
<td>Default Value</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------------</td>
<td>---------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td></td>
<td></td>
<td>and only if the workbooks come from a safe source and you have verified that they do not contain an unsafe expression.</td>
</tr>
<tr>
<td>vizqlserver.browser.render</td>
<td>true</td>
<td>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 About Client-Side Rendering for details.</td>
</tr>
<tr>
<td>vizqlserver.browser.render_</td>
<td>100</td>
<td>The default value (100) 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.</td>
</tr>
<tr>
<td>threshold</td>
<td></td>
<td></td>
</tr>
<tr>
<td>vizqlserver.browser.render_</td>
<td>60</td>
<td>The default value (60) represents a high level of complexity for a view displayed on a tablet. Complexity factors include number of marks, headers, reference</td>
</tr>
<tr>
<td>threshold_mobile</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Option</td>
<td>Default Value</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------------------------</td>
<td>---------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td></td>
<td></td>
<td>lines, and annotations. Views that exceed this level of complexity are rendered by the server instead of in the tablet's web browser.</td>
</tr>
<tr>
<td>vizqlserver.clear_session_on_unload</td>
<td>false</td>
<td>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. See General Performance Guidelines for more information.</td>
</tr>
<tr>
<td>vizqlserver.extsvc.connect_timeout_ms</td>
<td>1000</td>
<td>Extends the timeout value, in milliseconds, for connections to Microsoft’s RServer. Raise the value of this setting if Tableau is timing out before the server can respond.</td>
</tr>
<tr>
<td>vizqlserver.extsvc.host</td>
<td></td>
<td>Specifies an external service host.</td>
</tr>
</tbody>
</table>

**Note:** In versions of Tableau before version 10.1, this set-
<table>
<thead>
<tr>
<th>Option</th>
<th>Default Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>This setting, and the other vizqlserver.extsvc settings, support external service functionality in workbooks—in particular, R servers and Python servers. R is an open source software programming language and a software environment for statistical computing and graphics. In Tableau Desktop, you can use a set of four functions to pass R expressions to an Rserve server and obtain a result. If you upload a workbook that uses any of these functions, you should configure Tableau Server for an Rserve connection, by configuring this option and the three following. Otherwise, any worksheets that use R functionality will be unavailable.</td>
</tr>
<tr>
<td>Option</td>
<td>Default Value</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------</td>
<td>---------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>See <a href="https://help.tableau.com/current/pro/desktop/en-us/using_vizqlserver.htm">Pass Expressions to External Services</a> in the Tableau Help for further details.</strong></td>
</tr>
<tr>
<td>vizqlserver.extsvc.port</td>
<td>6311</td>
<td>Specifies an external service port. This setting supports R and Python functionality in workbooks.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Note:</strong> In versions of Tableau before version 10.1, this setting was named <code>vizqlserver.rserve.port</code>. Be sure to use this earlier setting name if your Tableau version is older than version 10.1.</td>
</tr>
<tr>
<td>vizqlserver.extsvc.username</td>
<td></td>
<td>Specifies an external service username. This setting supports R and Python functionality in workbooks. Not all Rserve hosts require a username and password.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Note:</strong> In versions of Tableau before version 10.1, this setting was named <code>vizqlserver.rserve.username</code>. Be</td>
</tr>
</tbody>
</table>

- 1195 -
<table>
<thead>
<tr>
<th>Option</th>
<th>Default Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>sure to use this earlier setting name if your Tableau version is older than version 10.1.</td>
</tr>
<tr>
<td>vizqlserv-era.extsvc.password</td>
<td></td>
<td>Specifies an external service password. This setting supports R and Python functionality in workbooks. Not all Rserve hosts require a username and password. <strong>Note</strong>: In versions of Tableau before version 10.1, this setting was named vizqlserv-era.rserve.password. Be sure to use this earlier setting name if your Tableau version is older than version 10.1.</td>
</tr>
<tr>
<td>vizqlserver.geosearch_cache_size</td>
<td>5</td>
<td>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</td>
</tr>
<tr>
<td>Option</td>
<td>Default Value</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------</td>
<td>---------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td></td>
<td></td>
<td>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 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).</td>
</tr>
</tbody>
</table>
| vizqlserver.log.level | info          | The logging level for vizqlserver Java components. Logs are written to ProgramData\Tableau\TableauServer-data\t-absvc\logs\vizqlserver\*.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. See Change Logging Levels for more information. |
<table>
<thead>
<tr>
<th>Option</th>
<th>Default Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>vizqlserver.port</td>
<td>9100</td>
<td>Base port for the VizQL servers.</td>
</tr>
<tr>
<td>vizqlserver.querylimit</td>
<td>1800</td>
<td>Longest allowable time for updating a view, in seconds.</td>
</tr>
<tr>
<td>vizqlserver.session.expiry.minimum</td>
<td>5</td>
<td>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.</td>
</tr>
<tr>
<td>vizqlserver.session.expiry.timeout</td>
<td>30</td>
<td>Number of minutes of idle time after which a VizQL session is discarded.</td>
</tr>
<tr>
<td>vizqlserver.showdownload</td>
<td>true</td>
<td>Controls the display of the Tableau Workbook option of the Download menu in views. When set to false, the Tableau Workbook option is unavailable.</td>
</tr>
<tr>
<td>vizqlserver.showshare</td>
<td>true</td>
<td>Controls the display of Share options in views. To hide these options, set to false.</td>
</tr>
<tr>
<td>Note: Users can override the server default by setting the &quot;showShareOptions&quot; JavaScript or URL parameter.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>vizqlserver.info</td>
<td>info</td>
<td>The logging level for trusted applications.</td>
</tr>
<tr>
<td>Option</td>
<td>Default Value</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------------</td>
<td>---------------</td>
<td>------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>er.trustedticket.log_level</td>
<td></td>
<td>authentication. The logs are written to <code>ProgramData\Tableau\TableauServer\data\t-\absvc\logs\vizqlserver\vizql-*.log</code>. Set to <code>debug</code> for more information. Using the debug level can significantly impact performance, so you should only use it when directed to do so by Tableau Support. See Change Logging Levels for more information.</td>
</tr>
<tr>
<td>vizqlserver.trustedticket.token_length</td>
<td>24</td>
<td>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. As of Tableau Server 10.5, this option is ignored unless <code>wgserv-er.trustedticket.use_deprecated_v2_tickets</code> is set to <code>true</code>, which is not a recommended best practice.</td>
</tr>
</tbody>
</table>
| vizqlserver.trustedticket.log_level | false         | When set to `true`, tickets are 9...
<table>
<thead>
<tr>
<th>Option</th>
<th>Default Value</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>er.trustedticket.use_deprecated_9digit_token</td>
<td>digits long (as in version 8.0 and earlier) and the setting vizqlserv-er.trustedticket.token_length is ignored.</td>
<td><strong>Warning:</strong> Setting this option to <code>true</code> severely and negatively impacts the security strength of trusted ticket authentication. For strongest security, upgrade to Tableau Server 10.5 and run trusted tickets in the default configuration.</td>
</tr>
<tr>
<td>vizqlserver.url_scheme_whitelist</td>
<td>Specifies one or more URL schemes to whitelist when using URL actions on views and dashboards. The schemes <code>http</code>, <code>https</code>, <code>gopher</code>, <code>mailto</code>, <code>news</code>, <code>sms</code>, <code>tel</code>, <code>tsc</code>, and <code>tsl</code> are whitelisted by default. This command can contain multiple comma and space-separated values enclosed by double quotes, as in this example:</td>
<td></td>
</tr>
<tr>
<td>Option</td>
<td>Default Value</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------------------------</td>
<td>---------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>tabadmin set vizqlserver.url_scheme_whitelist &quot;scheme1, scheme2&quot;</td>
<td></td>
<td>The values you specify overwrite previous settings (excluding the default settings). Therefore, you must include the full list of schemes in the <code>set</code> command. (You cannot amend the list of schemes by running the <code>set</code> command repeatedly.)</td>
</tr>
<tr>
<td>webdataconnector.enabled</td>
<td>true</td>
<td>When this setting is <code>true</code>, you can use <code>tabadmin</code> commands to manage web data connectors on the server, and web data connectors are included when you back up and restore the server. If the setting is <code>false</code>, web data connectors that are on the server are not included during backup and restore. For more information, see Web Data Connectors in Tableau Server.</td>
</tr>
<tr>
<td>web-data-connector.refresh.enabled</td>
<td>true</td>
<td>When this setting is <code>true</code>, the server supports doing refreshes for web data connector-based data sources. For more information, see Web Data Connectors in Tableau Server.</td>
</tr>
<tr>
<td>Option</td>
<td>Default Value</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------------------------</td>
<td>---------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>web-data-connector.whitelist.mode</td>
<td>mixed</td>
<td>Determines how Tableau Server can run web data connectors. Supported modes are:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- local. Users can run connectors that have been imported to Tableau Server.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- fixed. Users can run connectors that are on a safe list (whitelist) of URLs.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- mixed. Users can run imported connectors or connectors on the safe list.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- insecure. Users can run any connector.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Important:</strong> Use the insecure option <em>only</em> for development and testing.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Because connectors run custom code, running connectors that have not been vetted can pose a security threat.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>For more information about how to add connectors to a safe list and import connectors, see Web Data Connectors in Tableau</td>
</tr>
<tr>
<td>Option</td>
<td>Default Value</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------------------------</td>
<td>---------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>wgserver.audit_history_expiration_days</td>
<td>183</td>
<td>Specifies the number of days after which historical events records are removed from the PostgreSQL database (the Tableau Server database). See Collect Data with the Tableau Server Repository for details.</td>
</tr>
<tr>
<td>wgserver.authentication.desktop_nosaml</td>
<td>false</td>
<td>Controls whether or not Tableau Desktop uses SAML for authentication. Use this option when your IdP does not use forms-based authentication. Valid options are true and false. By default this is not set, so the behavior is equivalent to setting it to false. Set this to true to disable SAML authentication for Tableau Desktop.</td>
</tr>
<tr>
<td>wgserver.authentication.app_nosaml</td>
<td>false</td>
<td>Controls whether or not devices running older versions of Tableau Mobile may use SAML for authentication. Valid options are true and false. Devices running Tableau Mobile app version 19.225.1731 and higher ignore this option. To disable devices running Tableau Mobile...</td>
</tr>
<tr>
<td>Option</td>
<td>Default Value</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>---------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Default Value</td>
<td></td>
<td>Mobile app version 19.225.1731 and higher, disable SAML as a client login option on Tableau Server.</td>
</tr>
<tr>
<td>wgserv-er.authentication.login</td>
<td></td>
<td>In Tableau Server 10.2 and earlier, set to saml to enable SAML SSO authentication or set to openid to enable OpenID SSO Connect authentication.</td>
</tr>
<tr>
<td>wgserv-er.authentication.restricted</td>
<td>false</td>
<td>Controls whether users can sign in to Tableau Server using a Tableau Server username and password. This setting is useful in scenarios where users normally sign in to the server using single sign-on (OpenID Connect or Kerberos, for example). In these cases, if wgserv-er.authentication.restricted is set to true (the default is false), only system administrators can use tabcmd because this utility doesn’t support SSO so requires a username and password.</td>
</tr>
<tr>
<td>wgserver.change_owner.enabled</td>
<td>true</td>
<td>Controls whether the ownership of a workbook, data source or project can be changed. Other options include false and</td>
</tr>
<tr>
<td>Option</td>
<td>Default Value</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------------------------</td>
<td>----------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>adminonly</td>
<td></td>
<td>See Manage Content Ownership for details.</td>
</tr>
<tr>
<td>wgserver.clickjack_defense.enabled</td>
<td>true</td>
<td>When set to true, helps prevents a malicious person from &quot;clickjacking&quot; 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.</td>
</tr>
<tr>
<td>wgserver.domain.fqdn</td>
<td>value of %USERDOMAIN%</td>
<td>The fully qualified domain name of the Active Directory server to use.</td>
</tr>
<tr>
<td>wgserver.extended_trusted_ip_checking</td>
<td>false</td>
<td>Enforces IP client matching for trusted ticket requests.</td>
</tr>
<tr>
<td>wgserver.openid.iframed_idp.enabled</td>
<td>false</td>
<td>When enabled, if you are using embedded views and OpenID Connect, this suppresses the Tableau Server Sign In button and redirects the user to the IdP for authentication. This only works if the IdP does not implement clickjack pro-</td>
</tr>
<tr>
<td>Option</td>
<td>Default Value</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------------------------</td>
<td>---------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>wgserver.restrict_options_method</td>
<td>true</td>
<td>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.</td>
</tr>
<tr>
<td>wgserver.saml.enabled</td>
<td>false</td>
<td>In Tableau Server 10.3, set to true to enable SAML SSO authentication.</td>
</tr>
<tr>
<td>wgserver.saml.idpattribute.username</td>
<td>username</td>
<td>Specifies the name of the attribute in which your SAML IdP stores user names. By default, this is set to username. If the</td>
</tr>
<tr>
<td>Option</td>
<td>Default Value</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------------</td>
<td>---------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td></td>
<td></td>
<td>attribute name that your IdP uses contains spaces, enclose it in quotation marks. For more information, see Configure Server-Wide SAML or Configure Site-Specific SAML.</td>
</tr>
<tr>
<td>wgserver.saml.iframed_idp.enabled</td>
<td>false</td>
<td>When enabled, if you are using embedded views and SAML, this suppresses the Tableau Server Sign In button and redirects the user to the IdP for authentication.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>This only works if the IdP does not implement clickjack protection. If the IdP sign in page implements clickjack protection, the page will not display and the user cannot sign in. Most SAML IdPs implement clickjack protection and do not allow their sign in page to display in an &lt;iframe&gt; element.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>The default is false.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Important</strong>: Using this option disables Tableau Server clickjack protection for SAML, which can</td>
</tr>
<tr>
<td>Option</td>
<td>Default Value</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------------</td>
<td>--------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>wgserv-er.saml.logout.enabled</td>
<td>true</td>
<td>Specifies whether SAML logout is enabled for Tableau Server. The default is true. This setting only applies if SAML authentication is enabled for Tableau Server.</td>
</tr>
<tr>
<td>wgserv-er.saml.logout.redirect_url</td>
<td></td>
<td>Specifies the post-logout landing page for SAML authentication. The default is the standard server sign-in page. You can specify an absolute or a relative URL. For more information, see SAML Requirements.</td>
</tr>
<tr>
<td>wgserv-er.saml.maxassertiontime</td>
<td>3000</td>
<td>Specifies the maximum number of seconds, from creation, that an assertion is usable.</td>
</tr>
<tr>
<td>wgserv-er.sam-l.maxauthenticationage</td>
<td>7200</td>
<td>Specifies the maximum number of seconds allowed between user’s authentication and processing of the AuthNResponse message.</td>
</tr>
<tr>
<td>wgserv-er.saml.responseskew</td>
<td>180</td>
<td>Sets the maximum number of seconds difference between Tableau Server time and the time of the assertion creation</td>
</tr>
<tr>
<td>Option</td>
<td>Default Value</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------------------------------</td>
<td>---------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>wgserver.session.apply_lifetime_limit</td>
<td>false</td>
<td>Controls whether there is a session lifetime for server sessions. Set this to <code>true</code> to configure a server session lifetime.</td>
</tr>
<tr>
<td>wgserver.session.lifetime_limit</td>
<td>1440</td>
<td>The number of minutes a server session lasts if a session lifetime is set. The default is 1440 minutes (24 hours). If <code>wgserver.session.apply_lifetime_limit</code> is <code>false</code> (the default) this is ignored.</td>
</tr>
<tr>
<td>wgserver.session.idle_limit</td>
<td>240</td>
<td>The number of minutes of idle time before a sign-in to the web application times out.</td>
</tr>
<tr>
<td>wgserver.site_saml.enabled</td>
<td>false</td>
<td>Set to <code>true</code> to enable site-specific SAML so that each site on Tableau Server uses a different SAML identity provider (IdP).</td>
</tr>
<tr>
<td>wgserver.trusted_hosts</td>
<td></td>
<td>IP address or host names of web servers that request trusted tickets from Tableau Server. This command can contain multiple comma and space-separated values enclosed by double quotes, as in this example:</td>
</tr>
<tr>
<td>Option</td>
<td>Default Value</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------------------------</td>
<td>---------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>tabadmin set wgserver.trusted_hosts</td>
<td></td>
<td>The values you specify overwrite previous settings. Therefore, you must include the full list of hosts in the <code>set</code> command. (You cannot amend the list of hosts by running the <code>set</code> command repeatedly.) The web servers you specify must use static IP addresses, even if you use host names (learn more).</td>
</tr>
<tr>
<td>wgserver.trustedticket.use_deprecated_v2_tickets</td>
<td>false</td>
<td>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 10.5, 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 <code>true</code> if you have deployed trusted tickets with custom code that requires the legacy URL format. We recommend instead, updating</td>
</tr>
<tr>
<td>Option</td>
<td>Default Value</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>---------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>wgserver.unrestricted_ticket</td>
<td>false</td>
<td>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.</td>
</tr>
<tr>
<td>workerX.gateway.port</td>
<td>80 (443 if SSL)</td>
<td>External port that Apache listens on for workerX. worker0.gateway.port is Tableau Server's external port. In a distributed environment, worker0 is the primary Tableau Server.</td>
</tr>
<tr>
<td>workerX.vizqlserver.procs</td>
<td># of processes</td>
<td>Number of VizQL servers.</td>
</tr>
<tr>
<td>workerX.vizqlserver.port</td>
<td>9100</td>
<td>Base port for the vizQL server on workerX.</td>
</tr>
<tr>
<td>zoo-keeper.config.dataLogDir</td>
<td></td>
<td>Specifies the directory and file path for ZooKeeper transaction logs. By default ZooKeeper transaction logs are written to the Tableau data directory (for example c:\Tableau\Tableau Server-</td>
</tr>
</tbody>
</table>
### Option | Default Value | Description
--- | --- | ---

| | \data\t-absvc\zookeeper\0\data | Use this option to specify a different location. The drive and path apply to all nodes in a cluster. The location will be created if it does not exist. The drive must exist and be writable on all nodes. This should not be a UNC path to a share.

ZooKeeper recommends that transaction logs be written to a dedicated drive to optimize performance.

**Example:** `tabadmin set zoo-keep-er.config.dataLogDir "d:\Tableau\Tableau Server\zookeeper"

---

### Restore a Setting to its Default Value

You can restore the default value for a Tableau Server configuration setting by doing the following:

1. **Stop the server.**
2. Still in the bin directory, restore the default value for a particular setting by typing the
following:

tabadmin set option-name --default

For example, to set the tabadmin vizqlserver.session.expiry.timeout option back to its default value of 30 minutes, you would type the following:

tabadmin set vizqlserver.session.expiry.timeout --default

Alternatively, you can use the shorter -d command. For example:

tabadmin set vizqlserver.querylimit -d

3. Next, run the configure command:

tabadmin configure

4. Start the server.

**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.

**Note:** The tabcmd utility is included with Tableau Server, however its installer is not included. If you want to run it on a computer other than the primary server node, you need to download the installer from the Tableau website. For more information, see Install tabcmd below.

The tabcmd installer *is* included on the primary node with Tableau Server versions earlier than 10.1. If you have Tableau Server 10.0.x or earlier, you can find TabcmdInstaller.exe in `<server install location><version>\extras`. For example: D:\Tableau\Tableau Server-10.0\extras\TabcmdInstaller.exe.
In this article

- Install tabcmd
- How to use tabcmd
- Examples
- Status messages and logs

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.

**Note:** If you need the tabcmd installer for a version of Tableau Server prior to 10.1, it comes with your server installation. See the note above.

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.

5. Complete the installation steps as appropriate for the operating system of the tabcmd computer:

   • **Windows**

   Run the installer.

   We recommend that you install tabcmd to a folder named `tabcmd` at the root of the C:\ drive (C:\tabcmd), for convenience when using tabcmd, and to accommodate some limitations with the Windows operating system.

   **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.

   • **Linux**

   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, run the following command:

  sudo gdebi -i tableau-tabcmd-<version>_all.deb

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, change to the directory where tabcmd is installed.

   If you are using tabcmd on the Tableau Server on Windows primary node, change to the Tableau Server bin folder. For example:

   cd C:\Program Files\Tableau\Tableau Server\10.5\bin

   If you installed tabcmd on a computer other than the primary 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. If you are using tabcmd to perform more than one task, you must run tasks one after another (serially), rather than 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.
Status messages and logs

When a command is successful, tabcmd returns a status code of zero. A full error message for non-zero status codes is printed to stderr. In addition, informative or progress messages may be printed to stdout.

A full log named tabcmd.log that includes debugging, progress, and error messages is written to C:\Users\<username>\AppData\Local\Tableau.

tabcmd Commands

Looking for Tableau Server on Linux? See tabcmd Commands.

You can use the following commands with the tabcmd command line tool:

- addusers  (to group)
- creategroup
- createproject
- createsite
- createsiteusers
- createusers
- delete workbook-name or datasource-name
- deletegroup
- deleteproject
- deletesite
- deletesiteusers
- deleteusers
- editdomain
- editsite
- export
- get url
- initialuser
- listdomains
- listsites
- login
- logout
publish
publishsamples
refreshextracts
removeusers
runschedule
set
syncgroup
version

addusers group-name

Adds users to the specified group.

Example

tabcmd addusers "Development" --users "users.csv"

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.

If you use this command with large .csv files on Tableau Server, a server administrator can enable settings that help improve performance.

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 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
```

Top

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

  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
```

Top
createproject *project-name*

Creates a project.

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**

```
  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.

- `-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 site.

--storage-quota

In MB, the amount of workbooks, extracts, and data sources that can be stored on the site.

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
```

**Top**

`createsiteusers 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, role, 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 role, administrator level, and publisher permissions in the CSV file, you can pass role information to the command using the `--role` option.
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 assign the ServerAdministrator role to a user by using the createsiteusers command. (Use createusers instead.) If you specify the ServerAdministrator 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 role. However, if the server contains only one site (the default site), you can assign the ServerAdministrator role or specify system for the administrator value; in that case, the createsiteusers command works like the createusers command.

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.

Improving performance for large CSV files

A server administrator can use the tabadmin set command to enable settings that help to improve performance for large CSV files. For more information, see Improve performance for large CSV files passed through tabcmd in the CSV Import File Guidelines topic.

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 to the server. In that case, any password and friendly name information in the CSV file is ignored. Further, if a user is specified in the CSV file but there is no corresponding user in Active Directory, the user is not added to Tableau Server. For Active
Directory users, the user name is not guaranteed to be unique across domains, therefore you must include the domain as part of the user name (for example, example\Adam or adam@example.com.

While these can be sent either as domain/username or username@domain.com, we recommend using the domain/username format. See User Management in Active Directory Deployments for more information.

Example

tabcmd createsiteusers "users.csv" --role "Interactor"

Options

--admin-type

Deprecated. Use the --role option instead.

Use Site to assign or None to remove the site administrator right for any user who does not already have an administrator setting in the .csv file. The default is None for new users and unchanged for existing users. If the server contains multiple sites; system administrators cannot be created or demoted using createsiteusers. (Use createusers instead.)

--complete

Requires that all rows be valid for any change to succeed. This is the default setting.

--license

Deprecated. Use the --role option instead.

Specifies the license level (Interactor, Viewer, or Unlicensed) for any user who does not already have a license level setting in the .csv file. The default is Unlicensed for new users and unchanged for existing users.
**Note:** License levels were used in earlier versions of Tableau. They were replaced by site roles starting in Tableau Server 9.0.

--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.

Disallows publishing rights for any users who do not already have a publisher setting in the .csv file. This is a default value for new users.

--nowait

Do not wait for asynchronous jobs to complete.

--publisher

Deprecated. Use the **--role** option instead.

Assigns publishing rights for any user who does not already have a publisher setting in the .csv file. The default is no publishing rights (equivalent to **--no-publish**) for new users and unchanged for existing users.

-r, **--role**

Specifies a site role (ServerAdministrator, SiteAdministrator, Publisher, Interactor, ViewerWithPublish, Viewer, UnlicensedWithPublish, or Unlicensed) for any user who does not already have a role specified in the .csv file. The default is Unlicensed for new users and unchanged for existing users.
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.

**Note:** On Tableau Server, you cannot assign the **ServerAdministrator** role if the server has more than one site. In that case, use the **createuser** command.

**Note:** If you specify a role option, you cannot also include license, publisher, no-publisher, or administrator options.

**--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
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, role, 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 role, administrator level, and publisher permissions in the CSV file, you can pass role information to the command using the --role option.

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 to the server. In that case, any password and friendly name information in the CSV file is ignored. Further, if a user is specified in the CSV file but there is no corresponding user in Active Directory, the user is not added to Tableau Server. For Active Directory users, the user name is not guaranteed to be unique across domains, therefore you must include the domain as part of the user name (for example, example\Adam or adam@example.com).
While these can be sent either as domain/username or username@domain.com, we recommend using the domain/username format. See User Management in Active Directory Deployments for more information.

Example

```
tabcmd createusers "users.csv" --role "ServerAdministrator"

tabcmd createusers "users.csv"
```

Options

--admin-type

Deprecated. Use the --role option instead.

Assigns or removes the site administrator right (Site or None) for any user who does not already have an administrator setting in the .csv file. The default is None for new users and unchanged for existing users.

--complete

Requires that all rows be valid for any change to succeed. This is the default setting.

--license

Deprecated. Use the --role option instead.

Specifies the license level (Interactor, Viewer, or Unlicensed) for any user who does not already have a license level setting in the .csv file. The default is Unlicensed for new users and unchanged for existing users.

**Note:** License levels were used in earlier versions of Tableau Server, but have been replaced by site roles starting with Tableau Server 9.0.
--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.

Disallows publishing rights for any users who do not already have a publisher setting in the .csv file. This is a default value for new users.

--nowait

Do not wait for asynchronous jobs to complete.

--publisher

Deprecated. Use the --role option instead.

Assigns publishing rights for any user who does not already have a publisher setting in the .csv file. The default is no publishing rights (equivalent to --no-publish) for new users and unchanged for existing users.

-r, --role

Specifies a role (ServerAdministrator, SiteAdministrator, Publisher, Interactor, ViewerWithPublish, Viewer, UnlicensedWithPublish, or Unlicensed) for any user who does not already have a role specified in the .csv file. The default is Unlicensed for new users and unchanged for existing users.

On a multi-site server, the command does not assign the user to a site. Therefore, the only roles that the command will assign are ServerAdministrator and Unlicensed. In that case, if you specify a different role (like Publisher or Viewer), 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.

**Note:** If you specify a role option, you cannot also include license, publisher, no-publisher, or administrator options.

--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
```

Top
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.

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.

--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
```

**Top**

**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.

-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
```

**Top**

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

tabcmd deleteproject "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.

-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
```

Top

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

-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/S\text{heet}1` is a required value for the `export` command.

```
tabcmd export --csv -f "D:出口t0.csv" -- -430105/S\text{heet}1
```

**Top**

`deletesiteusers filename.csv`

Removes users from the site that you are logged into. 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 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"
```

Improving performance for large CSV files

A server administrator can use the `tabadmin set` command to enable settings that help to improve performance for large CSV files. For more information, see Improve per-
formance for large CSV files passed through tabcmd in the CSV Import File Guidelines topic.

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
```

**Top**

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`
  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
```

Top
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.

-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
```

Top
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.
--storage-quota

In MB, the amount of workbooks, extracts, and data sources that can be stored on the site.

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

Top

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 "workbook/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.

- **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 characters and PDF exports**: If you are exporting a view or workbook with a name that includes a character outside the ASCII character set, you need to URL encode (or 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**

```
  tabcmd export "Q1Sales/Sales_Report" --csv -f "Weekly-Report.csv"
```
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

Top
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

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"
tabcmd get "/views/Finance/InvestmentGrowth.png?:size=640,480" -f growth.png

tabcmd get "/views/Finance/InvestmentGrowth.png?:refresh=yes" -f growth.png

**Workbooks**

```
tabcmd get "/workbooks/Sales_Analysis.twb" -f "C:\Tableau_Workbooks\Weekly-Reports.twb"
```  

**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
```

**Top**

**initialuser**

Create the initial administrative user on a server that does not have an initial administrative user defined.

**Note:** The tabcmd initialuser command does not require authentication to Tableau Server, but you must run the command on the initial (primary) server node.

**Examples**
tabcmd initialuser --username "admin" --password "password" --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/Sble1 is a required value for the export command.

tabcmd export --csv -f "D:\export10.csv" -- -430105/Sble1

Top

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
```

Top

listsites

Returns a list of sites to which the logged in user belongs.

Example

```
tabcmd listsites --username adam --password mypassword
```

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
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 Content Permissions and Ownership.

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-certcheck -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

   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/Srel1 is a required value for the export command.

```
tabcmd export --csv -f "D:\export10.csv" -- -430105/Srel1
```

Top

logout

Logs out of the server.

Example

tabcmd logout

Top
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**

```bash
tabcmd publish "analysis.twbx" -n "Sales_Analysis" --db-username "jsmith" --db-password "secret-password"
```

```bash
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**

```bash
tabcmd publish "\\computer\volume\Tableau Workbooks\analysis.twbx" -n "Sales_Analysis" --db-username "jsmith" --db-password "secret-password"
```

```bash
tabcmd publish "\\computer\volume\Tableau Workbooks\analysis_sfdc.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.

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.

--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 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 --thumbnail-username option is set.

--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.

--append
Append the extract file to the existing data source.

--replace

Use the extract file to replace the existing data source.

--disable-uploader

Disable the incremental file uploader.

--restart

Restart the file upload.

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:
-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
publishsamples

Description

Publishes Tableau Sample workbooks to the specified project. Any existing samples will be overwritten.

Syntax

tabcmd publishsamples -n [project name] [Global options]

Example

Publish samples to the Inside Sales project on the Default site, as user jsmith.

tabcmd publishsamples -n "Inside Sales" -t "" -s localhost --username "jsmith" --password "secret-password"

Options

-n, --name

Required. Publishes the Tableau samples into the specified project. If the project name includes spaces, enclose the entire name in quotes.

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.

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

Top

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

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.

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.

--url

The name of the workbook as it appears in the URL. A workbook published as “Sales Analysis” has a URL name of “SalesAnalysis”.

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
```

**Top**

`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.

If you use this command with large `.csv` files on Tableau Server, a server administrator can enable settings that help to improve performance. For information, see Improve performance for large CSV files passed through `tabcmd`

`--[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:

- **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
```

Top

```
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
Top

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

Top
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**

**--administrator**

Deprecated. Use the **--role** option instead.

Assigns or removes the System or Site administrator right for users in the group. The **None** option removes the administrator right from all users in the group (except you, if you are a member of the group that you are synchronizing). If you do not include this option, users who are added to the group after you run the command are not assigned the administrator right.

**--license**

Deprecated. Use the **--role** option instead.
Specifies the license level (Interactor, Viewer, or Unlicensed) for users in the group.

**Note:** License levels were used in earlier versions of Tableau Server, but have been replaced by site roles starting in Tableau Server 9.0.

```
--no-publisher
```

Deprecated. Use the **--role** option instead.

Disallows publishing rights for users in the group.

```
--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.

Assigns publishing rights to users in the group.

```
-r, --role
```

Specifies a role (ServerAdministrator, SiteAdministrator, Publisher, Interactor, ViewerWithPublish, Viewer, UnlicensedWithPublish, or Unlicensed) for users in the group. The default is Unlicensed.
Note: If you specify a role option, you cannot also include license, publisher, no-publisher, or administrator options.

--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/Sample is a required value for the export command.

```
tabcmd export --csv -f "D:\export10.csv" -- -430105/Sample
```

**Top**

**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
```

Server Administrator Reference

Tableau Server Processes

Tableau Server installs a number of processes that work together to deliver the features that make up Tableau Server.


In this article

- Configuring processes
- Licensed processes
- List of processes

Configuring processes

Certain processes listed below cannot be configured: cluster controller and coordination service are installed on every node as part of the base install. They are required on every server node and do not count against a core-based license. Data engine is installed when you install file store, Vizportal, VizQL Server, data server, or backgrounder, and cannot be
installed separately. Every node that has an instance of one of those processes will also have a single instance of the data engine process.

The topics Performance Tuning Examples and High Availability describe some of the approaches you can take when configuring processes. High-level status for each process is displayed on the server's Status page and more detailed information related to some of the processes—such as the background process—is in the Administrative Views topic.

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 primary Tableau Server computer to verify they are licensed. If they cannot confirm there is a valid license, for example, if the primary 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.

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.

List of processes

For information on log files generated by these processes, see Server Log File Locations.

<table>
<thead>
<tr>
<th>Process</th>
<th>Purpose</th>
<th>Multi-Threading</th>
<th>Performance Characteristics</th>
<th>Licensed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Application Server</td>
<td>Handles</td>
<td>Yes</td>
<td>Only consumes noticeable resources</td>
<td>Yes</td>
</tr>
<tr>
<td>Process</td>
<td>Purpose</td>
<td>Multi-Threaded</td>
<td>Performance Characteristics</td>
<td>Licensed</td>
</tr>
<tr>
<td>--------------------------</td>
<td>-------------------------------------------------------------------------</td>
<td>----------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>----------</td>
</tr>
<tr>
<td>vizportal.exe</td>
<td>the web application, REST API-calls, supports browsing and searching</td>
<td></td>
<td>during infrequent operations, like publishing a workbook with an extract, or generating a static image for a view. Its load can be created by browser-based interaction and by tabcmd.</td>
<td></td>
</tr>
<tr>
<td>Backgrounder back-</td>
<td>Executes server tasks, including extract refreshes, subscriptions, 'Run Now' tasks, and tasks initiated from tabcmd</td>
<td>No</td>
<td>A single-threaded process where multiple processes can be run on any or all machines in the cluster to expand capacity. The backgrounder normally doesn't consume much process memory, but it can consume CPU, I/O, or network resources based on the nature of the workload presented to it. For example, performing large extract refreshes can use network bandwidth to retrieve data. CPU resources can be consumed by data retrieval or complex tabcmd tasks.</td>
<td>Yes</td>
</tr>
<tr>
<td>grounder.exe</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Cache Server redis-</td>
<td>Query cache</td>
<td>No</td>
<td>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 API server and applic-</td>
<td>No</td>
</tr>
<tr>
<td>server.exe</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Process</td>
<td>Purpose</td>
<td>Multi-Threaded</td>
<td>Performance Characteristics</td>
<td>Licensed</td>
</tr>
<tr>
<td>------------------</td>
<td>------------------------------------------------------------------------</td>
<td>----------------</td>
<td>---------------------------------------------------------------------------------------------</td>
<td>----------</td>
</tr>
<tr>
<td>Cluster Controller</td>
<td>Responsible for monitoring various components, detecting failures, and executing failover when needed</td>
<td>n/a</td>
<td>Included in the base install on every node.</td>
<td>No</td>
</tr>
<tr>
<td>zookeeper.exe</td>
<td>In distributed installations, responsible for ensuring there is a quorum for making decisions</td>
<td>n/a</td>
<td>Always installed on the primary node. For server installations with three to five nodes, also installed on the first two worker nodes. For server installations of more than five nodes, also installed on the first four worker nodes.</td>
<td>No</td>
</tr>
<tr>
<td>Process</td>
<td>Purpose</td>
<td>Multi-Threaded</td>
<td>Performance Characteristics</td>
<td>Licensed</td>
</tr>
<tr>
<td>---------------</td>
<td>--------------------------------</td>
<td>----------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>----------</td>
</tr>
<tr>
<td>Data Engine</td>
<td>Creates extracts and processes</td>
<td>Yes</td>
<td>The data engine's workload is generated by requests from the VizQL server, application server, API server, data server, and backgrounder server processes. The data engine services requests from most of the other server processes as well. It is the component that loads extracts into memory and performs queries against them, and writes the extracts to the database. The data engine is multi-threaded to handle multiple requests at a time. For more information about CPU and memory usage, see The New Data Engine in Tableau Server. The data engine is automatically installed when you install file store, VizQL Server, Application Server (VizPortal), Data Server, or Backgrounder. You cannot install Data Engine separately.</td>
<td></td>
</tr>
<tr>
<td>dataserver.exe</td>
<td>Manages connections to</td>
<td>Yes</td>
<td>Because it’s a proxy, it’s normally only bound by network, but it can be bound by CPU with enough simultaneous requests.</td>
<td>Yes</td>
</tr>
<tr>
<td>Process</td>
<td>Purpose</td>
<td>Multi-Threaded</td>
<td>Performance Characteristics</td>
<td>Licensed</td>
</tr>
<tr>
<td>-------------------------</td>
<td>------------------------------------------------------------------------</td>
<td>----------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>----------</td>
</tr>
<tr>
<td>Tableau Server data sources</td>
<td>user sessions. Its load is generated by browser- and Tableau Desktop-based interaction and extract refresh jobs for Tableau Server data sources.</td>
<td>Yes</td>
<td></td>
<td>No</td>
</tr>
<tr>
<td>File Store filestore.exe</td>
<td>automatically replicates extracts across data engine nodes</td>
<td>n/a</td>
<td>A data engine is automatically installed when file store is installed, unless the node already has an instance of data engine.</td>
<td>No</td>
</tr>
<tr>
<td>Repository postgres.exe</td>
<td>Tableau Server database, stores workbook and user metadata</td>
<td>n/a</td>
<td>Normally consumes few resources. It can become a bottleneck in rare cases for very large deployments (thousands of users) while performing operations such as viewing all workbooks by user or changing permissions. For more information, see Tableau Server Repository.</td>
<td>No</td>
</tr>
<tr>
<td>Search &amp; Browse server.exe</td>
<td>Handles fast search, filter, retrieval, and display of content metadata on the server</td>
<td>Yes</td>
<td>The process is memory bound first, and I/O bound second. The amount of memory used scales with the amount of content (number of sites/-projects/workbooks/datasources/views/users) on the server.</td>
<td>No</td>
</tr>
</tbody>
</table>
### Process | Purpose | Multi-Threaded | Performance Characteristics | Licensed
--- | --- | --- | --- | ---
VizQL Server | Loads and renders views, computes and executes queries | Yes | Consumes noticeable resources during view loading and interactive use from a web browser. Can be CPU bound, I/O bound, or network bound. Process load can only be created by browser-based interaction. Can run out of process memory. | Yes

## Tableau Server Ports

The following table lists the ports that Tableau Server uses by default, and which must be available for binding. If you install multiple instances of a process (Cache Server for example) on a node, consecutive ports are used, starting at the base port.

If Windows Firewall is enabled, Tableau Server will open the ports it needs for internal communication between processes. In some cases, if you are making an external connection to the Tableau Server database you may need to open ports manually. If have installed Tableau Server onto a Windows computer where Windows Firewall is disabled, then you will need to verify that the appropriate ports are open. In all cases, you should run Tableau Server on a computer where a firewall is enabled and configured to allow traffic only on ports specified in this topic.

### Dynamic port remapping

When dynamic port remapping is enabled (the default), Tableau Server first attempts to bind to the default ports, or to user-configured ports if they are defined. If the ports are not available, Tableau Server attempts to remap most processes to other ports, starting at port 8000. When next restarted, Tableau Server will revert to using the default or configured ports.
The gateway port and SSL port are not dynamically remapped. If port 80 is not available when Tableau Server is first installed, the installation program will choose a different gateway port (usually 8000). This value will display on the General tab of the Configuration utility. Tableau Server will always use the port shown in the Configuration utility for the gateway process.

When dynamic port remapping is disabled, Tableau Server does not attempt to remap processes and if a conflict is detected, Tableau Server will not start.

**Note:** Port conflicts can affect how JMX ports are determined. For more information, see Enable the JMX Ports.

You can disable dynamic port remapping using the `tabadmin set service.port_remapping.enabled` command. For more information, see `tabadmin` configuration options.

<table>
<thead>
<tr>
<th>Port</th>
<th>TCP/UDP</th>
<th>Used by ...</th>
<th>All</th>
<th>Distributed / High Availability</th>
<th>Parameter</th>
</tr>
</thead>
<tbody>
<tr>
<td>80</td>
<td>TCP</td>
<td>Gateway</td>
<td>X</td>
<td></td>
<td>gateway.public.port, workerX.gateway.port</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.</td>
<td>X</td>
<td></td>
<td>Tableau Server only supports SSL on port 443.</td>
</tr>
<tr>
<td>Port</td>
<td>TCP/UDP</td>
<td>Used by ...</td>
<td>TYPE OF INSTALLATION</td>
<td>Parameter</td>
<td></td>
</tr>
<tr>
<td>------</td>
<td>---------</td>
<td>-------------</td>
<td>----------------------</td>
<td>--------------------</td>
<td></td>
</tr>
<tr>
<td>2233</td>
<td>UDP</td>
<td>Server Resource Manager UDP port used for communication between Tableau Server processes. The Server Resource Manager monitors memory and CPU usage of Tableau Server processes (backgrounder.exe, dataserver.exe, tabprotsrv.exe, hyperd.exe, vizportal.exe, vizqlserver.exe).</td>
<td>X</td>
<td>resource_manager_port</td>
<td></td>
</tr>
<tr>
<td>3729</td>
<td>TCP</td>
<td>Tableau Server setup</td>
<td>X</td>
<td>--</td>
<td></td>
</tr>
<tr>
<td>3730–3731</td>
<td>TCP</td>
<td>Tableau worker servers in distributed and</td>
<td>X</td>
<td>--</td>
<td></td>
</tr>
<tr>
<td>Port</td>
<td>TCP/UDP</td>
<td>Used by ...</td>
<td>TYPE OF INSTALLATION</td>
<td>Parameter</td>
<td></td>
</tr>
<tr>
<td>-------</td>
<td>---------</td>
<td>--------------</td>
<td>-----------------------</td>
<td>-------------------------</td>
<td></td>
</tr>
<tr>
<td>5000</td>
<td>UDP</td>
<td>Server Worker Manager process (tabadmwrk.exe) that is used for auto-discovery of worker servers in a distributed environment.</td>
<td>All</td>
<td>Distributed / High Availability</td>
<td></td>
</tr>
<tr>
<td>6379</td>
<td>TCP</td>
<td>Cache Server process (redis-server.exe). Base port 6379. Consecutive ports after 6379 are used, up to the number of processes.</td>
<td>All</td>
<td>Distributed / High Availability</td>
<td>workerX.cacheserver.port</td>
</tr>
<tr>
<td>8060</td>
<td>TCP</td>
<td>PostgreSQL</td>
<td>All</td>
<td>Distributed / High Availability</td>
<td>pgsq1.port</td>
</tr>
<tr>
<td>Port</td>
<td>TCP/UDP</td>
<td>Used by ...</td>
<td>TYPE OF INSTALLATION</td>
<td>Parameter</td>
<td></td>
</tr>
<tr>
<td>------</td>
<td>---------</td>
<td>--------------</td>
<td>----------------------</td>
<td>-----------</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>All</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Distributed / High Availability</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Parameter</td>
<td></td>
</tr>
<tr>
<td>8061</td>
<td>TCP</td>
<td>PostgreSQL database. Used for verifying integrity of database for restoring.</td>
<td>X</td>
<td>pgsql.verify_restore.port</td>
<td></td>
</tr>
<tr>
<td>8062</td>
<td>TCP</td>
<td>PostgreSQL database</td>
<td>X</td>
<td>pgsqlX.port</td>
<td></td>
</tr>
<tr>
<td>8080</td>
<td>TCP</td>
<td>Solr, Tomcat HTTP, and Repository processes</td>
<td>X</td>
<td>solr.port, tomcat.http.port, repository.port</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>These parameters must be set to the same value.</td>
<td></td>
</tr>
<tr>
<td>8085</td>
<td>TCP</td>
<td>Tomcat HTTP</td>
<td>X</td>
<td>tomcat.server.port</td>
<td></td>
</tr>
<tr>
<td>8250</td>
<td>TCP</td>
<td>Background tasks</td>
<td>X</td>
<td>workerX.backgrounder.port</td>
<td></td>
</tr>
<tr>
<td>8350</td>
<td>TCP</td>
<td>Background tasks</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8600</td>
<td>TCP</td>
<td>Application Server process (vizportal.exe).</td>
<td>X</td>
<td>workerX.vizportal.port</td>
<td></td>
</tr>
<tr>
<td>Port</td>
<td>TCP/UDP</td>
<td>Used by ...</td>
<td>TYPE OF INSTALLATION</td>
<td>Parameter</td>
<td></td>
</tr>
<tr>
<td>--------</td>
<td>---------</td>
<td>-----------------------------------------------------------------------------</td>
<td>---------------------------------------</td>
<td>-------------------------</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>Base port 8600. Consecutive ports after 8600 are used, up to the number of processes.</td>
<td>Distributed / High Availability</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8700</td>
<td>TCP</td>
<td>Application Server process (vizportal.exe)</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>8755</td>
<td>TCP</td>
<td>Tableau Administrative process</td>
<td>X</td>
<td>tabadminservice.port</td>
<td></td>
</tr>
<tr>
<td>9100–</td>
<td>TCP</td>
<td>VizQL Server process (base port 9100). Consecutive ports after 9100, up to the number of processes, are also used. By default, Tableau Server installs with two VizQL Server processes (ports 9100 and 9101).</td>
<td>X</td>
<td>vizqlserver.port</td>
<td></td>
</tr>
<tr>
<td>9199</td>
<td>9101</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Port</td>
<td>TCP/UDP</td>
<td>Used by ...</td>
<td>TYPE OF INSTALLATION</td>
<td>Parameter</td>
<td></td>
</tr>
<tr>
<td>------------</td>
<td>---------</td>
<td>--------------------------------------</td>
<td>----------------------</td>
<td>-----------------</td>
<td></td>
</tr>
<tr>
<td>9200, 9400</td>
<td>TCP</td>
<td>VizQL Server process</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>9345</td>
<td>TCP</td>
<td>File Store service</td>
<td></td>
<td>filestore.port</td>
<td></td>
</tr>
<tr>
<td>9346</td>
<td>TCP</td>
<td>File Store status service</td>
<td></td>
<td>filestore.status.port</td>
<td></td>
</tr>
<tr>
<td>9700–9899</td>
<td>TCP</td>
<td>Data Server process (base port 9700). Consecutive ports after 9700, up to the number of processes, are also used. By default, Tableau Server installs with two Data Server processes (ports 9700 and 9701).</td>
<td></td>
<td>dataserver.port</td>
<td></td>
</tr>
<tr>
<td>9800, 10000</td>
<td>TCP</td>
<td>Data Server process</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>11000</td>
<td>TCP</td>
<td>Search server</td>
<td></td>
<td>workerX.searchserver.port</td>
<td></td>
</tr>
<tr>
<td>Port</td>
<td>TCP/UDP</td>
<td>Used by ...</td>
<td>TYPE OF INSTALLATION</td>
<td>Distributed / High Availability</td>
<td>Parameter</td>
</tr>
<tr>
<td>--------</td>
<td>---------</td>
<td>----------------------------------</td>
<td>----------------------</td>
<td>---------------------------------</td>
<td>------------------------------------------------</td>
</tr>
<tr>
<td>11100</td>
<td>TCP</td>
<td>Search server</td>
<td>X</td>
<td>All</td>
<td>workerX.search-server.startup.port</td>
</tr>
<tr>
<td>12000</td>
<td>TCP</td>
<td>Coordination controller (ZooKeeper) client port</td>
<td>X</td>
<td></td>
<td>workerX.zookeeper.port</td>
</tr>
<tr>
<td>12012</td>
<td>TCP</td>
<td>Cluster Controller process</td>
<td>X</td>
<td></td>
<td>cluster.status.port</td>
</tr>
<tr>
<td>12013</td>
<td>TCP</td>
<td>Cluster Controller process</td>
<td>X</td>
<td></td>
<td>cluster.storage.port</td>
</tr>
<tr>
<td>13000</td>
<td>TCP</td>
<td>Coordination controller (ZooKeeper) leader port</td>
<td>X</td>
<td></td>
<td>zookeeper.config.leaderPort</td>
</tr>
<tr>
<td>14000</td>
<td>TCP</td>
<td>Coordination controller (ZooKeeper) leader election port</td>
<td>X</td>
<td></td>
<td>zookeeper.config.leaderElectPort</td>
</tr>
<tr>
<td>2700-0</td>
<td>TCP</td>
<td>Workers and primary server to communicate licensing informations</td>
<td>X</td>
<td></td>
<td>--</td>
</tr>
<tr>
<td>Port</td>
<td>TCP/UDP</td>
<td>Used by ...</td>
<td>All</td>
<td>Type of Installation</td>
<td>Parameter</td>
</tr>
<tr>
<td>------</td>
<td>---------</td>
<td>-------------</td>
<td>-----</td>
<td>----------------------</td>
<td>-----------</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Distributed / High Availability</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TCP</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>7483</td>
<td>TCP</td>
<td>Data Engine process. Tableau Server installs with one Data Engine process per node where there are at least once instance of File Store, VizQL, Applic-</td>
<td>X</td>
<td></td>
<td>hyper.port</td>
</tr>
</tbody>
</table>
Edit the Default Ports

Tableau Server processes are configured to use certain ports on the computer where the server is installed. For more information, see Tableau Server Ports.

In general, you do not need to make changes to the port assignments for the server processes. However, if the computer that's running Tableau Server is also running other software that uses ports (this is not recommended), it's possible that the port assignments for Tableau Server processes conflict with ports used by the other software. In that case, you can assign different ports to Tableau Server processes.

To modify the ports used by Tableau Server processes, you use the command line administrative tool (tabadmin). For example, the default port for the application server process (vizportal.exe) is 8000. You can use the tabadmin parameter `workerX.vizportal.port` to change it to a different port.

**Note:** Changing ports requires a restart of Tableau Server. While the server is restarting it will be unavailable to all users. Be sure to warn your users of the outage prior to this operation or schedule this maintenance during non-business hours.
Follow the steps below to change the Tableau Server port configuration. If you are enabling the server's JMX ports, see Enable the JMX Ports

1. Open a command prompt as an administrator and type the following:
   
   ```
   cd "C:\Program Files\Tableau\Tableau Server\10.5\bin"
   ```

2. Modify a port value by typing one of the following commands:

   ```
   tabadmin set <workerX>.<parameter> <new port value>
   tabadmin set <parameter> <new port value>
   ```

   where:

   - `<workerX>` indicates which machine in a cluster you want to change the process port for. The placeholder `X` refers to the worker number—`worker0` is the primary server (or the only server if you are not running a distributed server), `worker1` is the first worker server, `worker2` is the second worker server, and so on. If you are running a distributed server and you want to change the default port for a process on all machines in the cluster, you need to run the command (from a command prompt on the primary) once for each machine in the cluster.

   - `<parameter>` is the server process that you are setting the port for, such as `vizportal.port`.

   - `<new port value>` is the new port number you want the server process to use.

Here’s an example that sets the port on the primary or standalone server to 8020 for the application server process (`vizportal`):

   ```
   tabadmin set worker0.vizportal.port 8020
   ```

The following example sets the port for a 3-machine cluster (one primary and two workers) to 9200 for the VizQL server process.

   ```
   tabadmin set worker0.vizqlserver.port 9200
   tabadmin set worker1.vizqlserver.port 9200
   ```
tabadmin set worker2.vizqlserver.port 9200

You can use the following parameters to modify the corresponding ports—see Tableau Server Ports for a complete list of tabadmin parameters that can be set.

<table>
<thead>
<tr>
<th>Port to Change</th>
<th>Parameter</th>
<th>Multiple workers?</th>
</tr>
</thead>
<tbody>
<tr>
<td>80</td>
<td>gateway.public.port</td>
<td>No</td>
</tr>
<tr>
<td>80</td>
<td>workerX.gateway.port</td>
<td>Yes</td>
</tr>
<tr>
<td>6379</td>
<td>workerX.cacheserver.port</td>
<td>Yes</td>
</tr>
<tr>
<td>8060</td>
<td>pgsql.port</td>
<td>Yes</td>
</tr>
<tr>
<td>8600</td>
<td>vizportal.port</td>
<td>Yes</td>
</tr>
<tr>
<td>9100</td>
<td>vizqlserver.port</td>
<td>Yes</td>
</tr>
<tr>
<td>9345</td>
<td>filestore.port</td>
<td>Yes</td>
</tr>
<tr>
<td>9700</td>
<td>workerX.dataserver.port</td>
<td>Yes</td>
</tr>
<tr>
<td>11000</td>
<td>workerX.searchserver.port</td>
<td>Yes</td>
</tr>
</tbody>
</table>

**Note:** You should not change port assignments for processes that are not listed in this table. Changing other ports can cause Tableau Server to stop working.

3. After you make the necessary port configuration changes, restart Tableau Server by typing the following:

    tabadmin restart

**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
   tabadmin set service.jmx_enabled true
   ```

3. Enter the configure command:
   
   ```bash
   tabadmin configure
   ```

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.

### JMX Port List

Here's the list of JMX ports, all of which are disabled by default. When these ports are enabled, they are used for all types of installations: single-server, distributed, and highly available:

<table>
<thead>
<tr>
<th>Port</th>
<th>Used by this server process ...</th>
<th>Parameter</th>
</tr>
</thead>
<tbody>
<tr>
<td>8550</td>
<td>Background monitor JMX. Determined by the background port of 8250 + 300.</td>
<td>--</td>
</tr>
<tr>
<td>8900 - 8959</td>
<td>Application server JMX. Determined by the application server port(s) + 300.</td>
<td>--</td>
</tr>
<tr>
<td>9095</td>
<td>Service monitor JMX.</td>
<td>svcmonitor.jmx.port</td>
</tr>
<tr>
<td>9400 - 9499</td>
<td>VizQL server JMX. Determined by the VizQL</td>
<td>--</td>
</tr>
</tbody>
</table>
How the JMX Ports Are Determined

By default, the JMX ports for the application server (8900 - 8959), backgrounder (8550), VizQL server (9400 - 9599), and the data server (10000 - 10299) are assigned using the formula “base port + 300”. (See Tableau Server Ports for a list of the default base ports.) In addition, if there are multiple instances of a process, each will have a JMX port. For example, if you configure Tableau Server to run four instances of the application server process, ports 8600 (default base port), 8601, 8602, and 8603 are used. Application server JMX ports 8900 (base port + 300), 8901, 8902, and 8903 are then bound to their respective process instances.

If dynamic port remapping is enabled (which is the default) and if a port conflict is detected, JMX ports are not determined using the "base port + 300" formula. Instead, both base ports and JMX ports are assigned to available ports starting at port 8000. No offset is used for JMX ports; they are assigned the next available port, just like base ports are. If it’s important that you have a fixed JMX port, you can disable port remapping or change the base ports so that there are no port conflicts.

Even though they’re not directly used by Tableau Server, if a JMX port is being used by another application, Tableau Server processes won’t run. In addition, JMX ports cannot be edited directly using tabadmin. You change a JMX port by changing the base port for its process. In other words, if port 10000 isn’t available for the data server JMX process, you use tabadmin (as described in Edit the Default Ports) to change the data server base port from 9700 to 9800. This will move the data server JMX port to 10100.

Restore the Default Value for a Port

You can restore the default value for a port by following the procedure below:

<table>
<thead>
<tr>
<th>Port</th>
<th>Used by this server process ...</th>
<th>Parameter</th>
</tr>
</thead>
<tbody>
<tr>
<td>server port(s) + 300.</td>
<td></td>
<td></td>
</tr>
<tr>
<td>10000 - 10299</td>
<td>Data server JMX. Determined by the data server port(s) + 300.</td>
<td>--</td>
</tr>
</tbody>
</table>
1. Open a command prompt as an administrator and type the following:

   \texttt{cd "C:\Program Files\Tableau\Tableau Server\10.5\bin"}

2. Restore the default port value by typing the following:

   \texttt{tabadmin set <workerX>.<parameter> --default}

   If Tableau Server is running on one machine, \texttt{<workerX> is worker0}. If you're running a cluster, \texttt{worker0 is the primary}, \texttt{worker1 is your first worker server}, \texttt{worker2 is your second}, and so on.

   Here's an example:

   \texttt{tabadmin set worker0.vizqlserver.port --default}

3. Update the server's configuration by typing the following:

   \texttt{tabadmin config}

4. Restart Tableau Server by typing the following:

   \texttt{tabadmin restart}

**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.

**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\jsmith`, where `example` is the NetBIOS name and `jsmith` 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 Dir-
ectory 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 users individually

In most case, Tableau Server uses the `sAMAccountName` value for the user name. When you import users individually from Active Directory (either by typing in their names or by using a CSV file), Tableau queries Active Directory with the user name that you provide. If a match is found, then that name is imported into Tableau Server and it becomes the name that the user enters in order to sign in to Tableau Server.

The user name that Tableau Server will import into the identity store will be the `sAMAccountName` value unless one of the following is true:

- If the user name that you specify is longer than 20 characters.
- If the user name that you specify contains an `@` character.

If the user name you enter meets either of these conditions, then Tableau will import the UPN prefix of the `userPrincipalName` attribute, which will become the user's Tableau logon user name.

If user names were inadvertently imported using UPN names, you can delete the accounts in Tableau Server and then reimport those accounts using the `sAMAccountName` value for the user name, as shown in **User logon name (pre-Windows 2000)** in the Windows **Active Directory Users and Computers** administrative console.

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.

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 user's 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:
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.

View Server Version

Server users can view the version of Tableau Server from the Help menu in the server web UI.

Viewing the server version from the Tableau Server web UI

- While logged into Tableau Server, click the information icon (i) and About Tableau Server.
The version of Tableau Server is listed in the About Tableau Server dialog box:

About Tableau Server

Tableau Server Version: 2019.1.0 (20191.18.1023.1646) 64-bit Windows
©2018 Tableau Software, Incorporated and its licensors. All rights reserved.

Close