Tableau Server för företag Driftsättningsguide

Senast uppdaterad 2025-02-10

© 2024 Salesforce, Inc.

Innehåll

Driftsättningsguide för Tableau Server för företag 1

Vem bör läsa detta? 1

Version 2

Framhävda funktioner 2

Licensiering 3

Del 1 – Förstå företagsdriftsättning 4

Industristandarder och driftsättningskrav 4

Säkerhetsåtgärder 5

Webbproxynivå 6

Belastningsutjämnare 6

Programnivå 7

Datanivå 7

Del 2 - Förstå Tableau Server driftsättning i referensarkitektur 8

Tableau Server-processer 8

PostgresSQL lagringsplats 10

Nod 1: Initial nod 10

Nod 1 reservomkoppling och automatisk återställning 11

Noderna 1 och 2: Applikationsservrar 11

Skalning av applikationsservrar 12

Noderna 3 och 4: Dataservrar 13

Skalning av dataservrar 13

Del 3 – förbereda för företagsdriftsättning av Tableau Server **15**

Undernät **16**

Gruppregler för brandväggar/säkerhet **16**

Webbnivå **16**

Programnivå **17**

Datanivå **17**

Bastion **18**

Exempel: Konfigurera undernät och säkerhetsgrupper i AWS **19**

AWS-referensarkitektur **20**

Bild 1: Topologin för VPC-undernät och EC2-instanser **20**

Bild 2: Protokollflöde och -anslutning **21**

Bild 3: Tillgänglighetszoner **22**

Bild 4: Säkerhetsgrupper **23**

AWS-tillgänglighetszoner och hög tillgänglighet **23**

VPC-konfiguration **23**

Konfigurera VPC **24**

Konfigurera säkerhetsgrupper **25**

Ange regler för inkommande och utgående trafik **26**

Gruppregler för offentlig säkerhet **26**

Gruppregler för privat säkerhet **26**

Gruppregler för data-säkerhet **27**

Gruppregler för bastion-värd-säkerhet **28**

Aktivera automatisk tilldelning av offentlig IP-adress ... 29
Belastningsutjämnare ... 29
Konfigurera värddatorer ... 30
Minsta rekommenderade hårdvara ... 30
Katalogstruktur ... 31
Exempel: Installera och förbereda värddatorer i AWS ... 31
Information om värd-instanserna ... 32
Tableau Server ... 32
Bastion-värd ... 32
Oberoende gateway för Tableau Server ... 32
PostgreSQL EC2-värd ... 33
Verifiering: VPC-anslutning ... 33
Exempel: Anslut till en bastion-värd i AWS ... 33
Del 4 – Installera och konfigurera Tableau Server ... 35
Innan du börjar ... 35
Installera, konfigurera och tar PostgreSQL ... 36
PostgreSQL-versionshantering ... 36
Installera PostgreSQL ... 38
Konfigurera Postgres ... 38
Skapa tar-säkerhetskopia (PostgreSQL steg 1) ... 39
Innan du installerar ... 41
Installera den ursprungliga noden för Tableau Server ... 41

Kör installationspaketet och initiera TSM 41
Aktivera och registrera Tableau Server 43
Konfigurera identitetslagret 44
Konfigurera extern Postgres 44
Avsluta installationen av nod 1 45
Verifiering: konfiguration av nod 1 46
Skapa tar-säkerhetskopior (steg 2) 47
Installera Tableau Server på kvarvarande noder 51
Generera, kopiera och använda startfilen för att initiera TSM 53
Konfigurera processer 53
Konfigurera nod 2 54
Konfigurera nod 3 55
Driftsätt samordningstjänstensemblen till nod 1–3 56
Skapa tar-säkerhetskopior (steg 3) 57
Konfigurera nod 4 61
Slutlig processkonfiguration och verifiering 61
Utför säkerhetskopiering 62
Del 5 – Konfigurera webbnivån 64
Oberoende gateway för Tableau Server 65
Autentisering och auktorisering 65
Förautentisering med en AuthN-modul 66
Konfigurationsöversikt 67

Exempel på webbnivåkonfiguration med oberoende gateway för Tableau Server 67
Förbereda miljön 68
Installera oberoende gateway 69
Oberoende gateway: direkt kontra omdirigerad anslutning 71
Konfigurera omdirigeringsanslutning 72
Konfigurera direkt anslutning 73
Verifiering: bastopologikonfiguration 74
Konfigurera lastbalanserare för AWS-program 75
Steg 1: Skapa målgrupp 75
Steg 2: Starta guiden för belastningsutjämnaren 76
Guidekonfiguration 76
Konfiguration för enskild sida 77
Steg 3: Aktivera varaktighet 78
Steg 4: Ställ in tidsgräns för inaktivitet för belastningsutjämnaren 79
Steg 5: Verifiera LBS-anslutning 79
Uppdatera DNS med den offentliga Tableau-URL:en 79
Verifiera anslutning 80
Exempel på autentiseringskonfiguration: SAML med extern IdP 80
Skapa ett Tableau-administratörskonto 80
Konfigurera Okta-program med förautentisering 81
Skapa och tilldela Okta-användare 83
Installera Mellon för förautentisering 83

Konfigurerar Mellon som förautentiseringsmodul ... 83
Skapa Tableau Server-applikation i Okta ... 86
Ställa in konfiguration av autentiseringsmodul på Tableau Server ... 86
Aktivera SAML på Tableau Server för IdP ... 87
Starta om tsig-httpd-tjänsten ... 89
Validera SAML-funktion ... 90
Konfigurera autentiseringsmodulen på den andra oberoende gateway-instansen ... 90
Del 6 - Konfigurera efter installation ... 93
Konfigurera SSL/TLS från belastningsutjämnare till Tableau Server ... 93
Innan du konfigurerar TLS ... 94
Konfigurera datorer med oberoende gateway för TLS ... 95
Steg 1: Distribuera certifikat och nycklar till dator med oberoende gateway ... 95
Steg 2: Uppdatera miljövariablerna för TLS ... 96
Steg 3: Uppdatera stubbkonfigurationsfilen för HK-protokollet ... 96
Steg 4: Kopiera stubbfil och starta om tjänsten ... 97
Konfigurera Tableau Server-nod 1 för TLS ... 97
Steg 1: Kopiera certifikat och nycklar och stoppa TSM ... 97
Steg 2: Ställ in certifikatresurser och aktivera oberoende gateway-konfiguration ... 98
Steg 3: Aktivera ”extern SSL” för Tableau Server och tillämpa ändringar ... 99
Steg 4: Uppdatera JSON-filen för gatewaykonfiguration och starta tsm ... 99
Uppdatera IdP-autentiseringsmodulens URL:er till HTTPS ... 100
Konfigurera AWS-belastningsutjämnare för HTTPS ... 100

Validera TLS ... 102
Konfigurera den andra instansen av oberoende gateway för SSL ... 102
Konfigurera SSL för Postgres ... 104
Valfritt: Aktivera certifikatförtroendevalidering på Tableau Server för Postgres SSL 106
Installera Postgres-klienten på nod 1 ... 107
Kopiera rotcertifikat till nod 1 ... 108
Ansluta till Postgres-värden via SSL från nod 1 ... 108
Konfigurera SMTP- och händelsemeddelanden ... 109
Installera PostgreSQL-drivrutin ... 110
Konfigurera stark lösenordspolicy ... 111
Del 7 – Validering, verktyg och felsökning ... 113
Validering av reservomkopplingssystem ... 113
Automatisk återställning av ursprunglig nod ... 114
Felsöka återställning av ursprunglig nod ... 116
Återskapa den misslyckade noden ... 116
switchto ... 116
Felsöka oberoende gateway för Tableau Server ... 119
Starta om tableau-tsig-tjänsten ... 119
Hitta felaktiga strängar ... 120
Söka i relevanta loggar ... 120
Loggfiler från oberoende gateway ... 120
Tableau Server tabadminagent-loggfil ... 121

Läsa in httpd-stubbfilen på nytt 122

Ta bort eller flytta loggfiler 122

Webbläsarfel 123

Verifiera TLS-anslutning från Tableau Server till oberoende gateway 124

Bilaga – AWS Deployment Toolbox 125

TabDeploy4EDG – automatiserat installationsskript 125

Exempel: Automatisera driftsättning av AWS-infrastruktur med Terraform 127

Mål 128

Sluttillstånd 128

Krav 130

Innan du börjar 130

Steg 1 – förbereda miljön 130

A. Ladda ner och installera Terraform: 130

B. Generera privata-offentliga nyckelpar 130

C. Ladda ner projektet och lägg till en tillståndskatalog 131

Steg 2: Anpassa Terraform-mallarna 131

versions.tf 132

key-pair.tf 132

locals.tf 132

providers.tf 132

elb.tf 133

variables.tf 134

modules/tableau_instance/ec2.tf 134
Steg 3 – kör Terraform 135
A. Initiera Terraform 135
B. Planera Terraform 135
C. Tillämpa Terraform 136
Valfritt: Förstör Terraform 136
Steg 4 – ansluta till Bastion 136
Steg 5 – Installera PostgreSQL 137
Steg 6 – (valfritt) kör DeployTab4EDG 138
Bilaga – Exempel på driftsättning på webbnivå med Apache 139
Installera Apache 140
Konfigurera proxy för att testa anslutning till Tableau Server 141
Verifiering: bastopologikonfiguration 142
Konfigurera lastbalansering på proxyn 142
Kopiera konfigurationen till den andra proxyservern 143
Konfigurera lastbalanserare för AWS-program 144
Steg 1: Skapa målgrupp 144
Steg 2: Starta guiden för belastningsutjämnaren 145
Guidekonfiguration 145
Konfiguration för enskild sida 146
Steg 3: Aktivera varaktighet 147
Steg 4: Ställ in tidsgräns för inaktivitet för belastningsutjämnaren 148

Steg 5: Verifiera LBS-anslutning ... 148
Uppdatera DNS med den offentliga Tableau-URL:en ... 148
Verifiera anslutning ... 148
Exempel på autentiseringskonfiguration: SAML med extern IdP ... 149
Skapa ett Tableau-administratörskonto ... 149
Konfigurera Okta-program med förautentisering ... 149
Skapa och tilldela Okta-användare ... 151
Installera Mellon för förautentisering ... 152
Konfigurerar Mellon som förautentiseringsmodul ... 152
Skapa Tableau Server-applikation i Okta ... 155
Aktivera SAML på Tableau Server för IdP ... 156
Validera SAML-funktion ... 158
Felsökning av validering ... 159
Konfigurera SSL/TLS från belastningsutjämnare till Tableau Server ... 160
Exempel: Konfigurera SSL/TLS i AWS-referensarkitektur ... 160
Steg 1: Ta reda på certifikat och relaterade nycklar ... 160
Steg 2: Konfigurera proxyserver för SSL ... 162
Steg 3: Konfigurera Tableau Server för extern SSL ... 164
Steg 4: Valfri autentiseringskonfiguration ... 165
Steg 5: Konfigurera AWS-lastbalanserare för HTTPS ... 165
Steg 6: Verifiera SSL ... 166

Driftsättningsguide för Tableau Server för företag

Guiden för driftsättning av Tableau Server för hela företaget (EDG) har utvecklats för att erbjuda föreskrivande vägledning gällande driftsättning av Tableau Server (lokalt eller i molnet). Guiden erbjuder vägledning gällande driftsättning för företags i samband med en referensarkitektur. Vi har testat referensarkitekturen för att verifiera efterlevnad med riktmärken gällande säkerhet, skala och prestanda som överensstämmer med branschstandardens bästa praxis.

På en hög nivå består de grundläggande funktionerna i en driftsättning för hela företaget, enligt branschstandard, av en nivåbaserad topologi där varje lager av serverapplikationens funktionalitet (webgatewaynivå, applikationsnivå och datanivå) är bunden och skyddad av åtkomstkontrollerade subnät. Användare som kommer åt serverapplikationen via internet autentiseras på webbnivån. När de har autentiserats skickas begäran till ett skyddat subnät där applikationsnivån hanterar företagslogiken. Data med högt värde skyddas av det tredje subnätet: datanivån. Tjänster på applikationsnivån kommunicerar över det skyddade nätverket till datanivån för att hantera dataförfrågningar till datakällor på servern.

Den här driftsättningen låter säkerheten ligga i framkant gällande alla designbeslut och implementeringar. Kontinuerlig drift, prestanda och skalbarhet är även de prioriterade krav. På grund av den distribuerade och modulära designen på referensarkitekturen kan kontinuerlig drift och prestanda skalas på ett linjärt och förutsägbart sätt genom att strategiskt samlokalisera kompatibla tjänster vid varje nod och lägga till tjänster vid flaskhalsar.

Vem bör läsa detta?

EDG har utvecklats för företags IT-administratörer som kan behöva

- en IT-hanterad Tableau-driftsättning.
- efterlevnad enligt branschstandarder.
- driftsättning enligt branschens bästa praxis.
- säker driftsättning som standard.

EDG är en implementeringsguide för driftsättning av arkitektur enligt företagsreferenser. Även om den här versionen av EDG innehåller ett exempel på AWS/Linux-implementering kan den användas som en resurs av erfarna IT-administratörer på företag för att driftsätta den föreskrivna referensarkitekturen enligt valfri branschstandard eller i valfri datacentermiljö.

Version

Den här versionen av EDG utvecklades för version 2021.2.3 (eller senare) av Tableau Server. Även om du kan använda EDG som en allmän referens för att driftsätta äldre versioner av Tableau Server rekommenderar vi att du driftsätter referensarkitekturen med Tableau Server 2021.2.3 eller senare. Vissa funktioner och alternativ är inte tillgängliga på äldre versioner av Tableau Server.

För de senaste funktionerna och alla förbättringar rekommenderar vi att du driftsätter EDG med Tableau Server 2022.1.7 eller senare.

Referensarkitekturen som beskrivs i den här guiden har stöd för följande Tableau-klienter: webbredigering med kompatibla webbläsare, Tableau Mobile och Tableau Desktop version 2021.2.1 eller senare. Andra Tableau-klienter (Tableau Prep och Bridge osv.) har ännu inte validerats med referensarkitekturen.

Framhävda funktioner

Den första versionen av referensarkitekturen för Tableau Server presenterar följande scenarier och funktioner:

- Förautentisering för klienter: Tableau-klienter (Desktop, Mobile, webbredigering) autentiseras med företagets autentiseringsleverantör på webbnivån innan de får åtkomst till den interna versionen av Tableau Server. Den här processen

hanteras genom att konfigurera ett authN-tillägg på den oberoende gatewayen för Tableau Server som agerar som den omvända proxyservern. Se Del 5 – Konfigurera webbnivån.

- Driftsättning med zero-trust: Då all trafik till Tableau-servrar redan är förautentiserad fungerar hela Tableau-driftsättningen under ett privat subnät som inte kräver en betrodd anslutning.
- Extern lagringsplats: Referensarkitekturen indikerar att Tableau-lagringsplatsen ska installeras på en extern PostgreSQL-databas vilket låter DBA:er hantera, optimera, skala och säkerhetskopiera lagringsplatsen som en generisk databas.
- Initial återställning av en nod: EDG introducerar ett skript som automatiserar den initial återställningen av en nod i händelse av ett fel.
- Tar-baserad säkerhetskopiering och återställning: Använd väl beprövade tar-säkerhetskopior vid strategiska delmål i Tableau-driftsättningen. Om driftsättningen skulle misslyckas eller konfigureras felaktigt kan du snabbt återställa till föregående steg i driftsättningen genom att återställa den tillhörande tar-säkerhetskopian.
- Prestandaförbättring: Kund- och labbvalidering visar en prestandaförbättring på 15–20 % när EDG körs jämfört med standardiserad driftsättning.

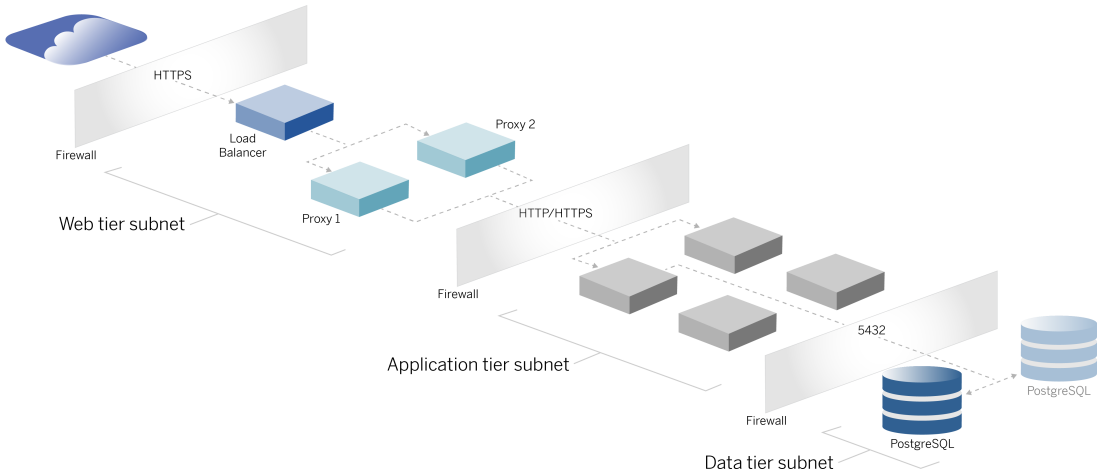
Licensiering

Referensarkitekturen som Tableau Server ska använda och som föreskrivs i den här guiden kräver en licens till Tableau Advanced Management för att aktivera den externa lagringsplatsen för Tableau Server. Du kan även valfritt driftsätta extern fillagring på Tableau Server, vilket även det kräver en licens till Tableau Advanced Management. Se *Om Tableau Advanced Management på Tableau Server* (Linux).

Del 1 - Förstå företagsdriftsättning

I del 1 beskrivs mer i detalj de funktioner och krav som är relaterade till industristandardiserad företagsdriftsättning och för vilka Driftsättningsguide för Tableau Server för företag har utformats.

Följande nätverksdiagram visar en generisk nivåindelad driftsättning av datacenter med Tableau Server-referensarkitektur.



Industristandarder och driftsättningskrav

Följande är egenskaper för driftsättning med industristandarder. Dessa är kraven som referensarkitekturen har utformats för:

- En nätverksdesign med flera nivåer: Nätverket är bundet av skyddade undernät för att begränsa åtkomsten vid varje skikt: webbskikt, applikationsskikt och dataskikt. Ingen enskild kommunikation kan passera över undernätet eftersom all kommunikation avslutas vid nästa undernät.
- Portar och protokoll blockerade som standard: Varje undernät eller säkerhetsgrupp kommersom standard att blockera alla inkommande och utgående portar och protokoll. Kommunikation aktiveras delvis genom att öppna undantag i port- eller protokollkonfigurationen.

- Medföljande webbautentisering: Användarbegäranden från internet autentiseras av en autentiseringsmodul i omvänd proxy på webbnivån. Därför autentiseras alla begäran till applikationsskiktet på webbnivån innan de övergår till det skyddade applikationsskiktet.
- Plattformsoberoende: Lösningen kan distribueras med lokala serverappar eller i molnet.
- Teknikagnostisk: Lösningen kan användas i en virtuell datormiljö eller i programbehållare. Den kan även distribueras på Windows eller Linux. Denna första version av referensarkitekturen och den stödjande dokumentationen har emellertid utvecklats för Linux som körs i AWS.
- Högt tillgänglig: Alla komponenter i systemet distribuerad som ett kluster och utformas för att fungera i en aktiv/aktiv eller aktiv/passiv driftsättning.
- Isolerade roller: Varje server utför en diskret roll. Denna design partitionerar alla servrar så att åtkomst kan minimeras till tjänstspecifika administratörer. DBA hanterar till exempel PostgreSQL för Tableau, identitetsadministratörer hanterar autentiseringsmodul på webbnivå, nätverks- och molnadministratörer aktiverar trafik och anslutningar.
- Linjärt skalbar: som diskreta roller kan du skala varje nivåtjänst oberoende av inläsningsprofil.
- Kundsupport: Referensarkitekturen stöder alla Tableau-klienter: Tableau Desktop (version 2021.2 eller senare), Tableau Mobile och Tableau webbredigering.

Säkerhetsåtgärder

Säkerhet är som tidigare nämnts en primär funktion i industristandardens datacenterdesign.

- Åtkomst: Varje nivå är bunden av ett undernät som framtvingar åtkomstkontroll i nätverkslagret med hjälp av portfiltrering. Kommunikationsåtkomst mellan undernäten kan också framtvingas av applikationsskiktet med autentiserade tjänster mellan processer.
- Integration: Arkitekturen är utformad för att anslutas till identitetsprovidern (IdP) I i omvänd proxy på webbnivån.
- Sekretess: Trafiken in på webbnivån krypteras från klienten med SSL. Trafiken in i de interna undernäten kan också eventuellt krypteras.

Webbproxynivå

Webbnivån är ett undernät i DMZ (även kallat perimeterzonen) som fungerar som en säkerhetsbuffert mellan internet och de interna undernäten där program används. Webbnivån är värd för omvända proxyservrar som inte lagrar några känsliga uppgifter. Omvända proxyservrar konfigureras med ett AuthN-plugin för att förhandsgodkänna klientsessioner med en betrodd IdP innan du omdirigerar klientbegäran till Tableau Server. Mer information finns i Förautentisering med en AuthN-modul.

Belastningsutjämnare

Distributionsdesignen omfattar en lösning för belastningsutjämnning för företag framför de omvända proxyservrarna.

Belastningsutjämnare ger viktiga säkerhets- och prestandaförbättringar genom att

- Virtualisera klientdels-URL:en för programmets nivåtjänster
- Framtvinga SSL-kryptering
- Avlasta SSL
- Framtvinga komprimering mellan klienten och webbnivåtjänsterna
- Skydda mot DOS-angrepp
- Tillhandahålla hög tillgänglighet

Obs! Oberoende gateway för Tableau Server ingår i Tableau Server version 2022.1. Oberoende gateway är en fristående instans av gatewayprocessen i Tableau som fungerar som en Tableau-medveten omvänd proxy. Vid lanseringen har Oberoende gateway validerats, men inte testats fullt ut i EDG-referensarkitekturen. När all testning är klar kommer EDG att uppdateras med instruktioner och riktlinjer för Oberoende gateway för Tableau Server.

Programnivå

Programnivån finns i ett undernät som kör serverprogrammets centrala affärslogik. Programnivån består av tjänster och processer som konfigureras över distribuerade noder i ett kluster. Det går endast att komma åt programnivån från webbnivån och den är inte direkt tillgänglig för användare.

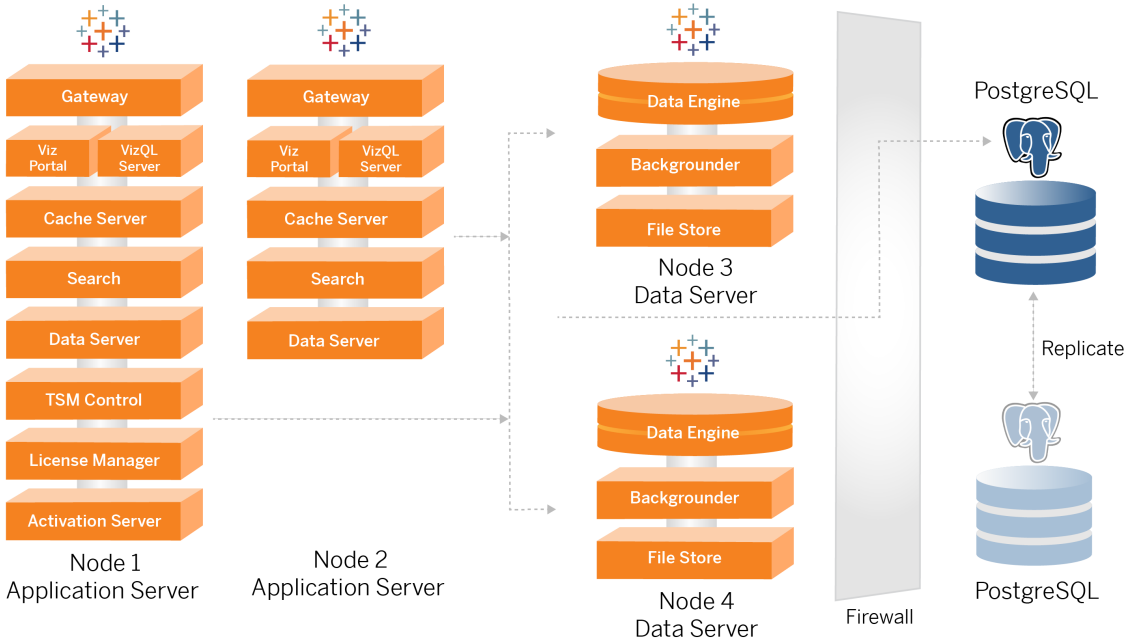
Prestanda och tillförlitlighet förbättras genom att konfigurera programprocesserna så att processer med olika resursanvändningsprofiler (dvs. CPU-intensiv jämfört med minnesintensiv) samlokaliseras.

Datanivå

Datanivån är ett undernät som innehåller värdefulla data. All trafik till denna nivå kommer från programnivån och är därför redan autentiserad. Förutom åtkomstkraven för nätverkslagret med portkonfiguration bör detta lager omfatta autentiserad åtkomst och eventuellt krypterad trafik med programnivån.

Del 2 - Förstå Tableau Server driftsättning i referensarkitektur

Följande bild visar de relevanta Tableau Server-processerna och hur de distribueras i referensarkitekturen. Denna driftsättning anses vara den minsta företagslämpliga Tableau Server-distributionen.



Processdiagrammen i detta ämne är avsedda att visa de viktigaste, definierande processerna för varje nod. Det finns många stödjande processer som också körs på noderna som inte visas i diagrammen. För en lista över alla processer, se konfigurationsavsnittet i denna guide, Del 4 – Installera och konfigurera Tableau Server.

Tableau Server-processer

Tableau Server-referensarkitekturen är en fyra noders Tableau Server-klusterdistribution med extern lagringsplats på PostgreSQL:

- Tableau Server initial nod (nod 1): Kör nödvändiga TSM-administrativa tjänster och licenstjänster som endast kan köras på en enda nod i klustret. I företagssammanhang är den initiala noden för Tableau Server den primära noden i klustret. Denna nod kör även redundanta applikationstjänster med nod 2.
- Tableau Server applikationsnoder (nod 1 och nod 2): De två noderna betjänar klientförfrågningar, ansluter till och söker datakällor och till datanoderna.
- Tableau Server datanoder (nod 3 och nod 4): Två noder som är dedikerade till att hantera data.
- Extern PostgreSQL: denna värd kör processen Tableau Server-lagringsplats. -För HA-driftsättning måste du köra ytterligare en PostgreSQL-värd för aktiv/passiv redundans.

  Du kan också köra PostgreSQL på Amazon RDS. Mer information om skillnaderna mellan att köra lagringsplatsen på en RDS-instans jämfört med en EC2-instans finns i *Extern lagringsplats för Tableau Server* (Linux).

  Distribuering av Tableau Server med en extern lagringsplats kräver en licens för Tableau Advanced Management.

  Om din organisation inte har intern DBA-expertis kan du eventuellt köra Tableau Server-lagringsplatsens process som standard, intern PostgreSQL-konfiguration. I standardscenariot körs lagringsplatsen på en Tableau-nod med inbäddad PostgreSQL. I detta fall rekommenderar vi att du kör lagringsplatsen på en dedikerad Tableau-nod, och ett passivt lagringsplatsen på en ytterligare dedikerad nod för att stödja reservomkoppling av lagringsplats. Se *Reservonkoppling av lagringsplats* (Linux).

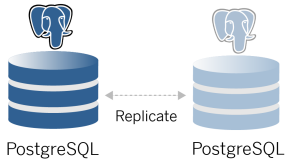
  Som exempel kan nämnas att den AWS-implementering som beskrivs i denna guide förklarar hur man distribuerar den externa lagringsplatsen på PostgreSQL som körs på en EC2-instans.

- Valfritt: Om din organisation använder extern lagring kan du distribuera Tableau fillagring som en extern tjänst. Den här guiden inkluderar inte det externa fillagret i kärnimplementeringsscenariot. Läs mer i *Installera Tableau Server med extern fillagring* (Linux).

För att driftsätta Tableau Server med en extern fillagring krävs en licens för Tableau Advanced Management.

PostgresSQL lagringsplats

Tableau Server-lagringsplats är en PostgresSQL-databas som lagrar serverdata. I dessa data ingår bland annat Tableau Server-relaterad information om användare, grupper, grupptilldelningar, behörigheter, projekt, datakällor, extraktmetadata och uppdateringsinformation.



Den förvalda PostgresSQL-distributionen förbrukar nästan 50 % av systemets minnesresurser. Baserat på dess användning (för produktion och stora produktionsdistributioner) kan resursanvändningen öka. Av den anledningen rekommenderar vi att du kör lagringsprocessen på en dator som inte kör några andra resursintensiva serverkomponenter som VizQL, bakgrundsprocessor eller datamotor. Att köra lagringsplatsprocessen tillsammans med någon av dessa komponenter kommer att skapa IO-konflikter, resursbegränsning och försämra den totala prestandan för distributionen.

Nod 1: Initial nod

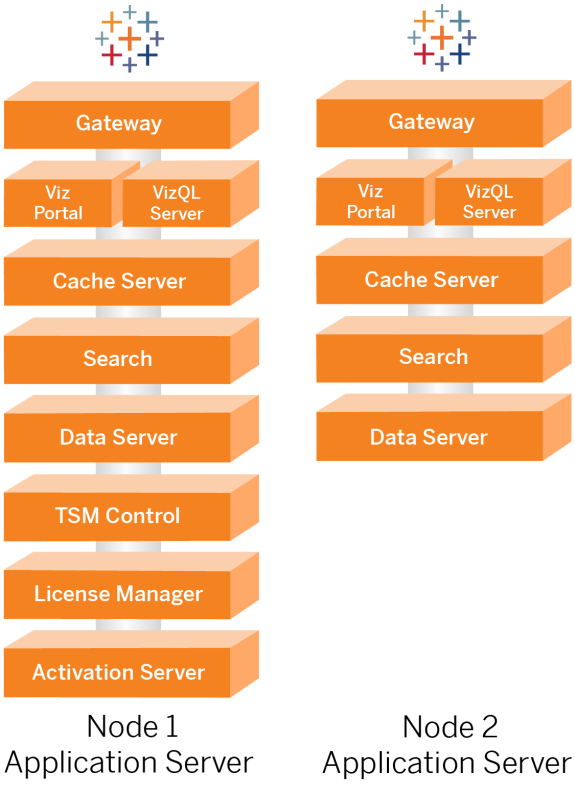
Den initiala noden kör ett litet antal viktiga processer och delar applikationsbelastningen med nod 2.

Den första datorn som du installerar Tableau på, den ”ursprungliga noden”, har vissa unika egenskaper. Det finns tre processer som bara körs på den ursprungliga noden och som inte kan flyttas till en annan nod förutom i händelse av ett fel: Licenstjänst (Licenshanteraren), Aktiveringstjänst och TSM-styrenhet (Administrationsstyrenhet).

Nod 1 reservomkoppling och automatisk återställning

Licensen, aktiveringen och TSM-styrenheten är avgörande för hur bra en Tableau Server-driftsättning fungerar. I händelse av ett nod 1-fel kommer användarna fortfarande att kunna ansluta till Tableau Server-distributionen, eftersom en korrekt konfigurerad referensarkitektur kommer att dirigera förfrågningar till nod 2. Utan dessa huvudtjänster kommer dock distributionen att befinna sig i ett kritiskt tillstånd av pågående fel. Se Automatisk återställning av ursprunglig nod.

Noderna 1 och 2: Applikationsservrar



Noderna 1 och 2 kör Tableau Server-processerna som betjänar klientförfrågningar, frågedatakällor, genererar visualiseringar, hanterar innehåll och administration och annan kärnverksamhetslogik i Tableau. Programservrarna lagrar inte användardata.

Obs! ”Application Server” är en term som också hänvisar till en Tableau Server-process som listas i TSM. Den underliggande processen för ”Application Server” är VizPortal.

Kör parallellt, nod 1 och nod 2 skalar upp till serviceförfrågningar från logiken för belastningsutjämning som körs på omvända proxyservrar. I egenskap av redundanta noder, hanteras klientförfrågningar och service av den återstående noden om en av dessa noder skulle sluta att fungera.

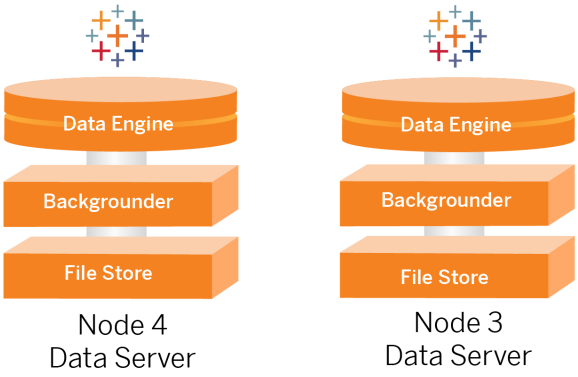
Referensarkitekturen har utformats så att kostnadsfria applikationsprocesser körs på samma dator. Detta innebär att processerna inte konkurrerar om datorresurser och skapar konflikter.

Till exempel är VizQL, en kärnbehandlingstjänst på applikationsservrar, mycket CPU- och minnesbunden. Dessutom använder VizQL nästan 60-70 % av CPU och minne på datorn. Av denna anledning är referensarkitekturen utformad så att inget annat minne eller processorbundna processer är på samma nod som VizQL. Testning visar att mängden av belastning eller antalet användare inte påverkar minnet eller CPU-användningen på VizQL-noder. Att till exempel minska antalet samtidiga användare i vårt belastningstest påverkar endast kontrollpanelens prestanda eller laddningsprocessen för visualisering, men det minskar inte resursutnyttjandet. Baserat på det tillgängliga minnet och CPU under toppanvändning kan du därför överväga att lägga till fler VizQL-processer. Som en utgångspunkt för typiska arbetsböcker, tilldela 4 kärnor per VizQL-process.

Skalning av applikationsservrar

Referensarkitekturen är utformad för skala upp baserat på en användningsbaserad modell. Som en allmän utgångspunkt rekommenderar vi minst två applikationsservrar som var och en stöder upp till 1000 användare. I takt med att användarbasen ökar ska du planera att lägga till en applikationsserver för varje ytterligare 1000 användare. Övervaka användning och prestanda för att justera användarbasen per värd för din organisation.

Noderna 3 och 4: Dataservrar



Processerna fillagring, datamotor (Hyper) och bakgrundsprocessor är samlokaliserade på noderna 3 och 4 av följande skäl:

- Extraktoptimering: Genom att köra bakgrundsprocessorn, Hyper och fillagring på samma nod optimeras prestanda och tillförlitlighet. Under extraheringsprocessen frågar bakgrundsprocessorn efter måldatabasen, skapar Hyper-filen på samma nod och laddar sedan upp till fillagringsplatsen. Genom att samlokalisera dessa processer på samma nod kräver arbetsflödet för skapande av extraktion inte kopiering av datamängder över nätverket eller noderna.
- Kostnadsfri resursutjämning: Bakgrundsprocessorn är huvudsakligen CPU-intensiv. Datamotor är en minnesintensiv process. Koppling av dessa processer möjliggör maximalt resursutnyttjande på varje nod.
- Konsolidering av dataprocesser: Eftersom var och en av dessa processer är backend-dataprocesser, är det logiskt att köra dem på den säkraste datanivån. I framtida versioner av referensarkitekturen kommer applikations- och dataserverna att köras på separata nivåer. På grund av applikationsberoende i Tableau-arkitekturen måste dock applikations- och dataserverna köras på samma nivå vid denna tidpunkt.

Skalning av dataservrar

Precis som med applikationsservrar kräver planering av de resurser som krävs för Tableau-dataservrar användarbaserad modellering. Anta i allmänhet att varje dataserver kan stödja upp till 2000 jobb för uppdatering av extrakt per dag. När dina extraktjobb ökar lägger du till ytterligare dataservrar utan fillagringstjänsten. Generellt sett är distributionen av dataservern

med två noder lämplig för driftsättningar som använder det lokala filsystemet för fillagringstjänsten. Observera att om du lägger till fler applikationsservrar påverkas inte prestandan eller skalan på dataserverna på ett linjärt sätt. Med undantag för vissa allmänna omkostnader från ytterligare användarfrågor, är effekten av att lägga till fler programvärdar och användare i själva verket minimal.

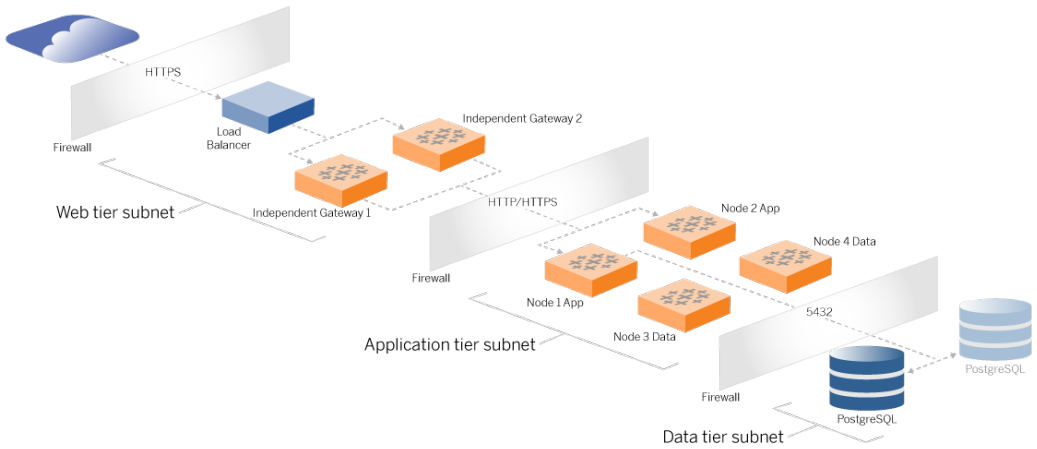
Del 3 – förbereda för företagsdriftsättning av Tableau Server

Del 3 beskriver kraven som finns för att förbereda infrastrukturen för att driftsätta Tableau Server-referensarkitekturen. Innan du börjar rekommenderar vi att du läser, Del 2 - Förstå Tableau Server driftsättning i referensarkitektur.

Förutom beskrivningar av kraven erbjuder detta avsnitt ett exempel på implementering av referensarkitekturen i en AWS-miljö. Resten av den här guiden bygger på exemplet med AWS-referensarkitekturen som påbörjades i detta avsnitt.

En grundläggande princip för referensarkitekturen är standardisering med bästa praxis för säkerheten i datacenter. Det är arkitekturen som är specifikt designad för att segregera tjänster i skyddade undernät inom nätverket. Kommunikationen mellan subnät är begränsad till specifika protokoll och trafik vid specifika portar.

Följande diagram visar designen på referensarkitekturens undernät för en lokal driftsättning eller en kundhanterad molnbaserad driftsättning. Du kan hitta ett exempel på molnbaserad driftsättning i avsnittet nedan, Exempel: Konfigurera undernät och säkerhetsgrupper i AWS.



Undernät

Skapa tre undernät:

- En webbnivå
- En applikationsnivå
- Ett undernät för data.

Gruppregler för brandväggar/säkerhet

Flikarna nedan beskriver reglerna gällande brandväggar för varje nivå i datacentret. För gruppregler gällande AWS-specifik säkerhet kan du läsa avsnittet längre fram i det här ämnet.

Webbnivå

Webbnivån är ett offentligt DMZ-undernät som hanterar inkommande HTTPS-förfrågningar och skicka vidare förfrågningarna till applikationsnivån. En sådan design erbjuder ett lager med försvar mot malware som kan riktas mot organisationen. Webbnivån blockerar åtkomst till applikationen/datanivån.

Trafik	Typ	Protokoll	Portintervall	Källa

Inkommande	SSH	TCP	22	Bastion-undernät (för molndriftsättningar)
Inkommande	HTTP	TCP	80	Internet (0.0.0.0/0)
Inkommande	HTTPS	TCP	443	Internet (0.0.0.0/0)
Utgående	All trafik	Allt	Allt	

Programnivå

Applikationens undernät befinner sig där Tableau Server-driftsättningen finns. Applikationens undernät inkluderar Tableau-programservrarna (nod 1 och nod 2). Tableau-applikationsservrarna bearbetar användarförfrågningar till dataservrarna och kör verksamhetens grundläggande logik.

Applikationens undernät inkluderar även Tableau-dataservrarna (nod 3 och nod 4).

All klienttrafik till applikationsnivån autentiseras på webbnivån. Administrativ åtkomst till applikationens undernät autentiseras och dirigeras genom bastion-värden.

Trafik	**Typ**	**Protokoll**	**Portintervall**	**Källa**
Inkommande	SSH	TCP	22	Bastion-undernät (för molndriftsättningar)
Inkommande	HTTPS	TCP	443	Undernät på webbnivå
Utgående	All trafik	Allt	Allt	

Datanivå

Dataundernätet finns där PostgreSQL-databasservern finns.

Trafik	Typ	Protokoll	Portintervall	Källa
Inkommande	SSH	TCP	22	Bastion-undernät (för molndriftsättningar)
Inkommande	PostgreSQL	TCP	5432	Undernät på applikationsnivå
Utgående	All trafik	Allt	Allt	

Bastion

De flesta säkerhetsteamen i företagsklass tillåter inte direkt kommunikation från det lokala administrativa systemet till noderna som är placerade i molnet. Istället fördelas all administrativ SSH-trafik till molnnoderna via en bastion-värd (även kallad en ”hoppserver”). För molnbaserade driftsättningar rekommenderar vi proxyanslutningar via en bastion-värd till alla resurser i referensarkitekturen. Detta är en valfri konfiguration för lokala miljöer.

Bastion-värden autentiserar administrativ åtkomst och tillåter endast trafik över SSH-protokollet.

Trafik	Typ	Protokoll	Portintervall	Källa	Mål
Inkommande	SSH	TCP	22	IP-adressen till administratörens dator	
Utgående	SSH	TCP	22		Undernät på webbnivå
Utgående	SSH	TCP	22		Undernät på applikationsnivå

Exempel: Konfigurera undernät och säkerhetsgrupper i AWS

Det här avsnittet beskriver steg för steg-procedurer till att skapa och konfigurera VPC- och nätverksmiljön för driftsättningen av Tableau Server-referensarkitekturen i AWS.

Bilderna nedan visar referensarkitekturen över fyra lager. Allt eftersom du ser över bilderna placeras komponentelementen på topologikartan i lager:

1. Topologin för VPC-undernät och EC2-instanser: en bastion-värd, två omvända proxyservrar, fyra Tableau-servrar och minst en PostgreSQL-server.
2. Protokollflöde och internetanslutning. All inkommande trafik hanteras via en AWS-internetgateway. Trafik till internet dirigeras via den NAT som används.
3. Tillgänglighetszoner Proxy-, Tableau Server- och PostgreSQL-värdarna är jämnt driftsatta över två tillgänglighetszoner.
4. Säkerhetsgrupper. Fyra säkerhetsgrupper (Offentlig, Privat, Data och Bastion) skyddar varje nivå på protokollnivån.

AWS-referensarkitektur

Bild 1: Topologin för VPC-undernät och EC2-instanser

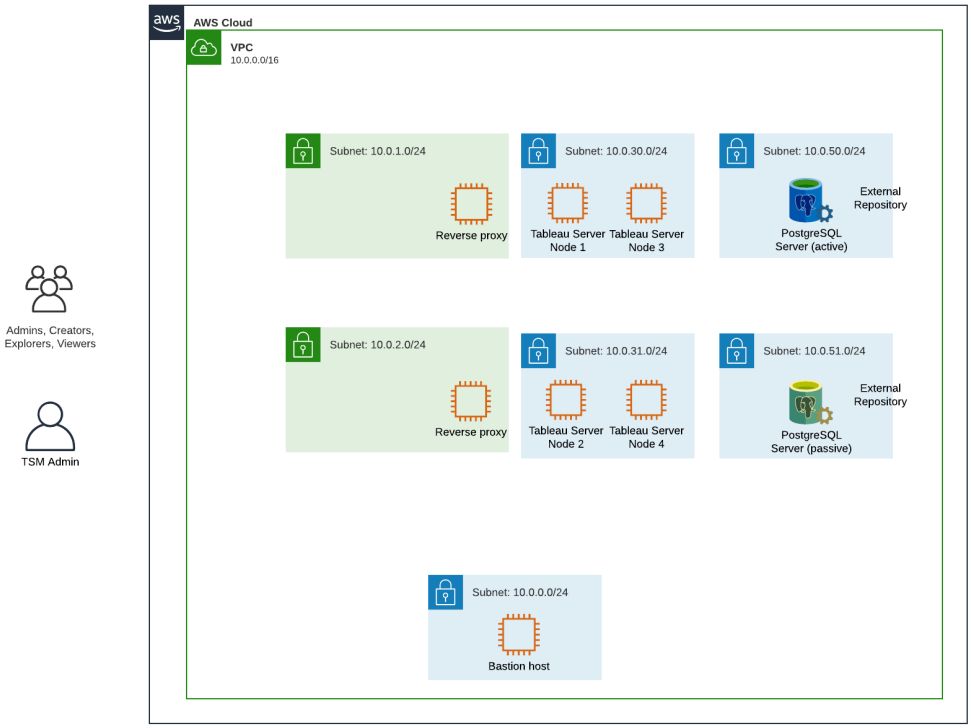


Bild 2: Protokollflöde och -anslutning

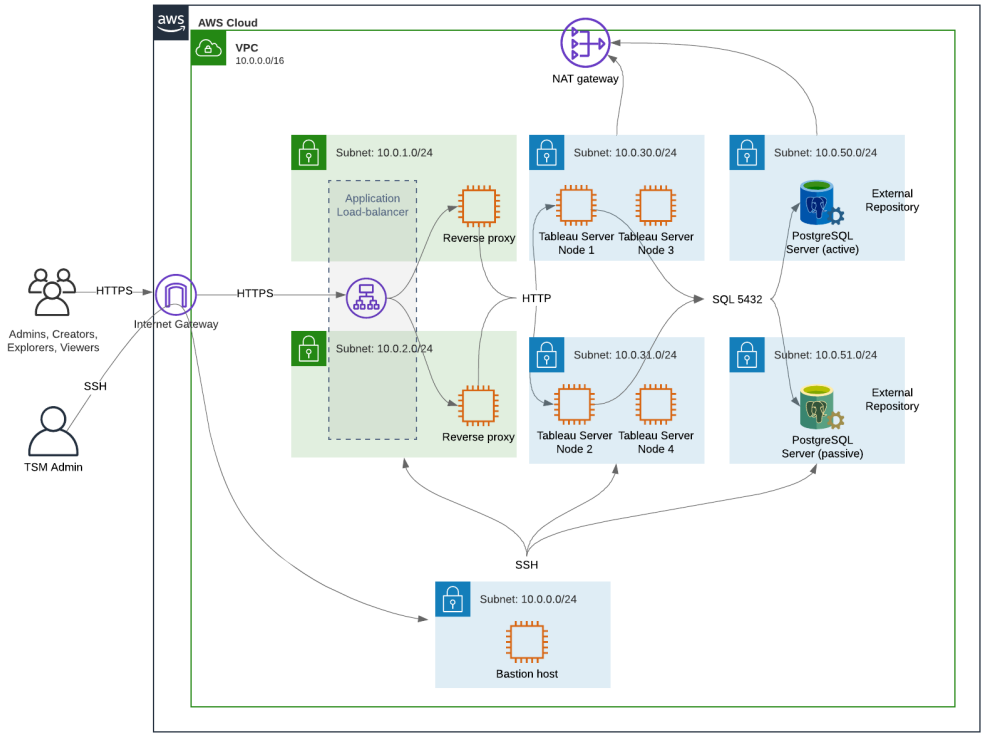


Bild 3: Tillgänglighetszoner

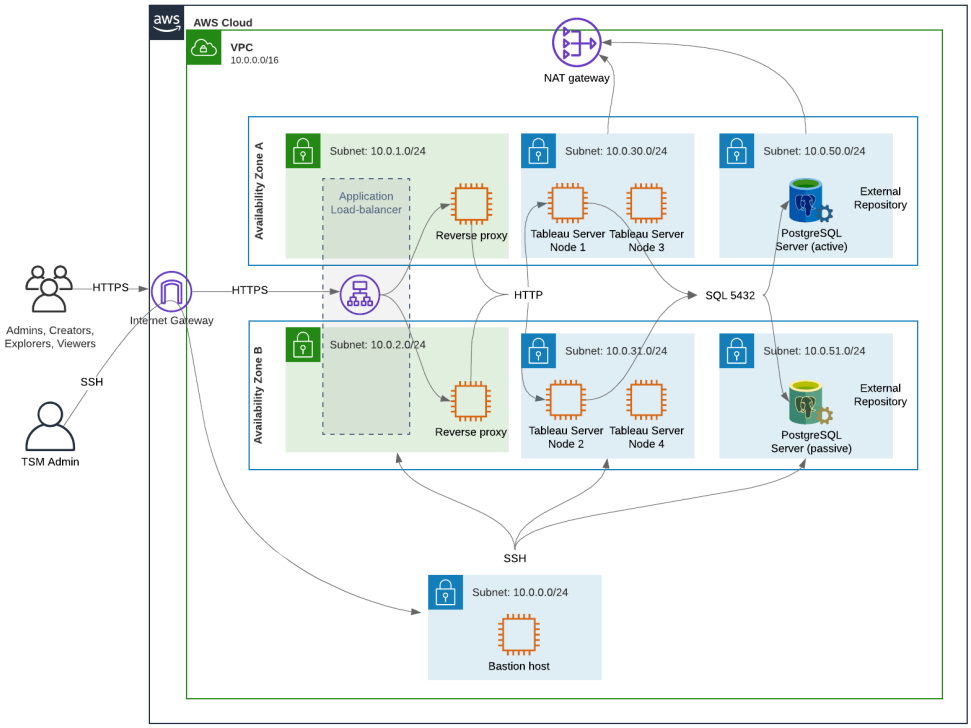
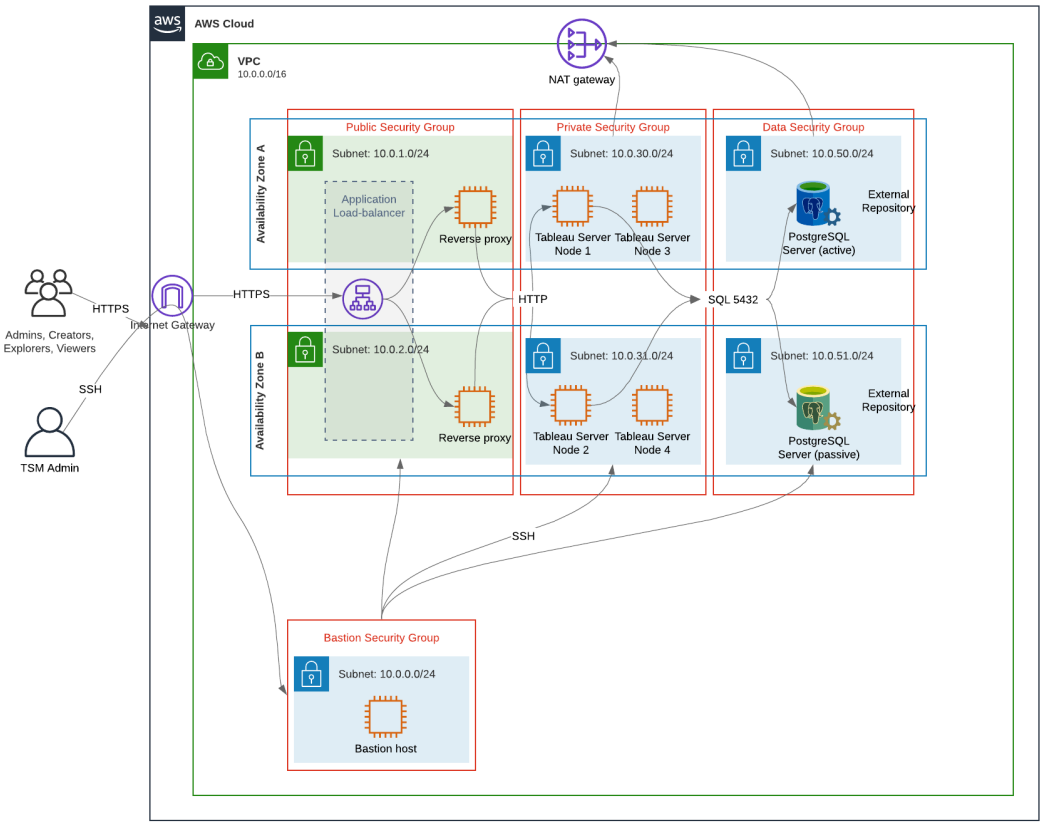


Bild 4: Säkerhetsgrupper



AWS-tillgänglighetszoner och hög tillgänglighet

Referensarkitekturen som visas i den här guiden använder sig av en driftsättning som erbjuder tillgänglighet via redundans när en enskild värd slutar fungera. I instansen där AWS-referensarkitekturen är driftsatt över två tillgänglighetszoner äventyras tillgängligheten, i det mycket sällsynta fallet, då en tillgänglighetszon slutar fungera.

VPC-konfiguration

Det här avsnittet beskriver hur du:

- Installera och konfigurera en VPC
- Konfigurerar internetanslutningen

- Konfigurerar undernät
- Skapar och konfigurerar säkerhetsgrupper

Konfigurera VPC

Proceduren i det här avsnittet mappas till användargränssnittet i den ”klassiska” VPC Experience. Du kan växla så att användargränssnittet visar den klassiska vyn genom att stänga av New VPC Experience i det övre vänstra hörnet av AWS VPC Dashboard.

Kör VPC-guiden för att skapa standardiserade privata och offentliga undernät samt standardiserad routing- och nätverks-ACL.

1. Du måste skapa en elastisk IP-adress innan du kan konfigurera en VPC. Skapa en allokering med alla standardinställningar.
2. Kör VPC-guiden > ”VPC med offentliga och privata undernät”
3. Acceptera de flesta standardinställningar. Förutom följande:
   - Ange ett VPC-namn.
   - Ange den elastisk IP-adressens allokerings-ID.
   - Ange följande CIDR-masker:
     - Offentliga undernät med IPv4 CIDR: 10.0.1.0/24, byt namn på detta undernät till `Public-a`.
     - Privata undernät med IPv4 CIDR: 10.0.30.0/24, byt namn på detta undernät till `Private-a`.
   - Tillgänglighetszon: för båda undernäten ska du välja alternativet **a** för den region du befinner dig i.

     > **Obs**! I det här exemplet använder vi **a** och **b** för att skilja mellan tillgänglighetszoner i ett givet AWS-datacenter. I AWS kanske namnen på tillgänglighetszonerna inte matchar exemplen som visas här. Vissa tillgänglighetszoner inkluderar till exempel **c**- och **d**-zoner i ett datacenter.
4. Klicka på **Skapa VPC**.
5. När en VPC har skapats ska du skapa undernäten `Public-b`, `Private-b`, `Data` och `Bastion`. Klicka på **Undernät** > **Skapa undernät** för att skapa ett undernät.

- `Public-b`: För tillgänglighetszon ska du välja alternativet **b** för den region du befinner dig i. CIDR-block: 10.0.2.0/24
- `Private-b`: För tillgänglighetszon ska du välja alternativet **b** för den region du befinner dig i. CIDR-block: 10.0.31.0/24
- `Data`: För tillgänglighetszon ska du välja zon **a** för den region du befinner dig i. CIDR-block: 10.0.50.0/24 Valfritt: Om du planerar att replikera den externa databasen över ett PostgreSQL-kluster ska du skapa ett Data-b-undernät i tillgänglighetszon b med ett CIDR-block om 10.0.51.0/24.
- `Bastion`: Välj endera zon för tillgänglighetszonen. CIDR-block: 10.0.0.0/24

6. När undernäten har skapats kan du redigera dirigeringstabellerna på de offentliga- och Bastion-undernäten för att använda de som är konfigurerad för vår associerade internetgateway (IGW). Samt redigera undernäten Offentlig och Data för att använda dirigeringstabellen som är konfigurerad för nätverkets adressöversättare (NAT).
   - Klicka på **Dirigeringstabeller** i AWS-kontrollpanelen för att avgöra vilken dirigeringstabell som är konfigurerad med IGW eller NAT. Välj en av de två länkade dirigeringstabellerna för att öppna sidan Egenskaper. Se målvärdet vid **Rutter** > **Destination** > **0.0.0.0/0**. Målvärdet skiljer på typen av rutt och börjar antingen med strängen `igw-` eller `nat-`.
   - För att uppdatera dirigeringstabellerna, **VPC** > **Undernät** > [subnet_name] > **Dirigeringstabell** > **Redigera dirigeringstabellens association**.

Konfigurera säkerhetsgrupper

VPC-guiden skapar en enda säkerhetsgrupp som inte kommer att användas. Skapa följande säkerhetsgrupper (**Säkerhetsgrupper** > **Skapa säkerhetsgrupp**). EC2-värdarna installeras i dessa grupper över två olika tillgänglighetszoner såsom visas på bilddiagrammet ovan.

- Skapa en ny säkerhetsgrupp: **Privat**. Det är här som alla fyra noder i Tableau Server installeras. Senare under installationsprocessen kopplas den privata säkerhetsgruppen till undernäten 10.0.30.0/24 och 10.0.31.0/24.
- Skapa en ny säkerhetsgrupp: **Offentlig**. Det är här som proxyservrar installeras. Senare under installationsprocessen kopplas den offentliga säkerhetsgruppen till undernäten 10.0.1.0/24 och 10.0.2.0/24.
- Skapa en ny säkerhetsgrupp: **Data**. Det är här som PostgreSQL med en extern Tableau-lagringsplats installeras. Senare under installationsprocessen kopplas den data-säkerhetsgruppen till undernätet 10.0.50.0/24 (och alternativt 10.0.51.0/24).

- Skapa en ny säkerhetsgrupp: **Bastion**. Det är här som bastion-värden ska installeras. Senare under installationsprocessen kopplas bastion-säkerhetsgruppen till undernätet 10.0.0.0/24.

Ange regler för inkommande och utgående trafik

I AWS är säkerhetsgrupper analoga med brandväggar i en lokal miljö. Du måste ange typen av trafik (såsom https, https, osv.), protokoll (TCP eller UDP) och portar eller portintervall (t.ex. 80, 443, osv.) som får komma in och/eller ut ur säkerhetsgruppen. För varje protokoll måste du även ange destinationen eller källtrafiken.

Gruppregler för offentlig säkerhet

Regler för inkommande trafik			
Typ	**Protokoll**	**Portintervall**	**Källa**
HTTP	TCP	80	0.0.0.0/0
HTTPS	TCP	443	0.0.0.0/0
SSH	TCP	22	Bastion-säkerhetsgrupp

Regler för utgående trafik			
Typ	**Protokoll**	**Portintervall**	**Mål**
All trafik	Allt	Allt	0.0.0.0/0

Gruppregler för privat säkerhet

Den privata säkerhetsgruppen inkluderar en regel för inkommande trafik för att tillåta HTTP-trafik från den offentliga säkerhetsgruppen. Tillåt HTTP-trafik endast under driftsättningen för att verifiera anslutningen. Vi rekommenderar att HTTP-regeln tas bort för inkommande trafik efter att den omvända proxyn har installerats och SSL har konfigurerats till Tableau.

Regler för inkommande trafik			
Typ	**Protokoll**	**Portintervall**	**Källa**
HTTP	TCP	80	Offentlig säkerhetsgrupp
HTTPS	TCP	443	Offentlig säkerhetsgrupp
PostgreSQL	TCP	5432	Datasäkerhetsgrupp
SSH	TCP	22	Bastion-säkerhetsgrupp
All trafik	Allt	Allt	Privat säkerhetsgrupp

Regler för utgående trafik			
Typ	**Protokoll**	**Portintervall**	**Mål**
All trafik	Allt	Allt	0.0.0.0/0
PostgreSQL	TCP	5432	Datasäkerhetsgrupp
SSH	TCP	22	Bastion-säkerhetsgrupp

Gruppregler för data-säkerhet

Regler för inkommande trafik			
Typ	**Protokoll**	**Portintervall**	**Källa**
PostgreSQL	TCP	5432	Privat säkerhetsgrupp
SSH	TCP	22	Bastion-säkerhetsgrupp

Regler för utgående trafik			
Typ	**Protokoll**	**Portintervall**	**Mål**

All trafik	Allt	Allt	0.0.0.0/0
PostgreSQL	TCP	5432	Privat säkerhetsgrupp
SSH	TCP	22	Bastion-säkerhetsgrupp

Gruppregler för bastion-värd-säkerhet

Regler för inkommande trafik			
Typ	**Protokoll**	**Portintervall**	**Källa**
SSH	TCP	22	IP-adressen och nätmasken för den dator som du ska använda för att logga in på AWS (administratörsdatorn).
SSH	TCP	22	Privat säkerhetsgrupp
SSH	TCP	22	Offentlig säkerhetsgrupp

Regler för utgående trafik			
Typ	**Protokoll**	**Portintervall**	**Mål**
SSH	TCP	22	IP-adressen och nätmasken för den dator som du ska använda för att logga in på AWS (administratörsdatorn).
SSH	TCP	22	Privat säkerhetsgrupp
SSH	TCP	22	Offentlig säkerhetsgrupp
SSH	TCP	22	Datasäkerhetsgrupp
HTTPS	TCP	443	0.0.0.0/0 (Valfritt: skapa den här regeln om du behöver ha åtkomst

			till internet för att ladda ner programvara som har stöd för bastion-värden)

Aktivera automatisk tilldelning av offentlig IP-adress

Detta skapar en IP-adress som låter dig ansluta till proxyservrarna och bastion-värden.

För offentliga- och bastion-undernät:

1. Välj undernätet
2. Under menyn **Åtgärder** ska du välja ”Ändra inställningar för automatisk tilldelning av IP-adress”.
3. Klicka på ”Aktivera automatisk tilldelning av offentliga IPv4-adresser”.
4. Klicka på **Spara**.

Belastningsutjämnare

Obs! Om du installerar i AWS och följer exemplet med driftsättningen, i den här guiden, bör du installera och konfigurera AWS-belastningsutjämnaren senare under driftsättningen. Detta sker enligt beskrivningen i Del 5 – Konfigurera webbnivån.

För lokala driftsättningar bör du kontakta nätverksadministratörerna för att driftsätta belastningsutjämnare som stödjer webbnivån för referensarkitekturen:

- En belastningsutjämnare för webborienterade program som accepterar HTTPS-begäranden från Tableau-klienter och som kommunicerar med de omvända proxyservrarna.
- Omvänd proxy:
  - Vi rekommenderar minst två proxyservrar för redundans och för att hantera klientbelastningen.
  - Tar emot HTTPS-trafik från belastningsutjämnaren.
  - Stöder ihållande sessioner mot Tableau-värden.

    - Konfigurera proxy för belastningsutjämning med resursallokering (round robin) för varje Tableau Server som kör gatewayprocessen.
    - Hanterar autentiseringsbegäranden från en extern identitetsprovider (IdP).
- Proxy för vidarebefordran: Tableau Server kräver åtkomst till internet för licensierings- och kartfunktioner. Beroende på miljön där din proxy för vidarebefordran är konfigurerad kan du behöva konfigurera dess godkännandelistor för Tableau Service-URL:er. Läs mer i *Kommunicera med Internet* (Linux).

Konfigurera värddatorer

Minsta rekommenderade hårdvara

Följande rekommendationer baseras på våra tester av verkliga data i referensarkitekturen.

Programservrar:

- CPU: 8 fysiska kärnor (16 virtuella processorer)
- RAM: 128 GB (16 GB/fysisk kärna)
- Diskutrymme: 100 GB

Dataservrar

- CPU: 8 fysiska kärnor (16 virtuella processorer)
- RAM: 128 GB (16 GB/fysisk kärna)
- Diskutrymme: 1 TB. Om driftsättningen ska använda extern lagring för fillagring i Tableau måste du beräkna det lämpliga diskutrymmet. Läs mer i *Installera Tableau Server med extern fillagring* (Linux).

Proxyservrar

- CPU: 2 fysiska kärnor (4 virtuella processorer)
- RAM: 8 GB (4 GB/fysisk kärna)
- Diskutrymme: 100 GB

Externa lagringsplatser för databasen

- CPU: 8 fysiska kärnor (16 virtuella processorer)
- RAM: 128 GB (16 GB/fysisk kärna)
- Kravet på diskutrymme beror på databelastningen och hur den påverkar säkerhetskopieringen. Läs mer i *Säkerhetskopierings- och återställningsprocesser* i avsnittet *Diskutrymmeskrav* (Linux).

Katalogstruktur

Referensarkitekturen rekommenderar Tableau Server-paketet och data installeras på icke-standardplatser:

- Installera paketet på: `/app/tableau_server`: Skapa den här katalogsökvägen innan Tableau Server-paketet installeras och ange den sedan under installationen.
- Installera Tableau-data till: `/data/tableau_data`. Skapa inte den här katalogen innan du installerar Tableau Server. Du måste istället ange sökvägen under installationen. Tableau-installationen skapar och ger sedan behörigheter till sökvägen på ett lämpligt sätt.

Se Kör installationspaketet och initiera TSM för information om implementeringen

Exempel: Installera och förbereda värddatorer i AWS

Det här avsnittet förklarar hur du installerar EC2-värdar för varje servertyp i Tableau Server-referensarkitekturen.

Referensarkitekturen kräver åtta värdar:

- Fyra instanser för Tableau Server.
- Två instanser för proxyservrar (Apache).
- Ett exempel för bastion-värden.
- En eller två EC2 PostgreSQL-databasinstanser

Information om värd-instanserna

Installera värddatorer enligt informationen nedan.

Tableau Server

- Amazon Linux 2
- Instanstyp: m5a.8xlarge
- Säkerhetsgrupp-ID: Privat
- Lagring: EBS, 150 GiB, gp2-volymtyp. Om driftsättningen ska använda extern lagring för fillagring i Tableau måste du beräkna det lämpliga diskutrymmet. Läs mer i *Installera Tableau Server med extern fillagring* (Linux).
- Nätverk: installera två EC2-värdar i varje privata undernät (10.0.30.0/24 och 10.0.31.0/24).
- Kopiera den senaste underhållsversionen av Tableau Server 2021.2 (eller senare) med rpm-paketet, från nedladdningssidan för Tableau, till varje Tableau-värd.

Bastion-värd

- Amazon Linux 2
- Instanstyp: t3.micro
- Säkerhetsgrupp-ID: Bastion
- Lagring: EBS, 50 GiB, gp2-volymtyp
- Nätverk: Bastion-undernät 10.0.0.0/24

Oberoende gateway för Tableau Server

- Amazon Linux 2
- Instanstyp: t3.xlarge
- Säkerhetsgrupp-ID: Offentligt
- Lagring: EBS, 100 GiB, gp2-volymtyp
- Nätverk: Installera en EC2-instans i varje offentligt undernät (10.0.1.0/24 och 10.0.2.0/24)

PostgreSQL EC2-värd

- Amazon Linux 2
- Instanstyp: r5.4xlarge
- Säkerhetsgrupp-ID: Data
- Lagring: Kravet på diskutrymme beror på databelastningen och hur den påverkar säkerhetskopieringen. Läs mer i *Säkerhetskopierings- och återställningsprocesser* i avsnittet *Diskutrymmeskrav* (Linux).
- Nätverk: data-undernät 10.0.50.0/24. (Om du replikerar PostgreSQL i ett HA-kluster ska du sedan installera den andra värden i undernätet 10.0.51.0/24)

Verifiering: VPC-anslutning

Verifiera nätverkskonfigurationen efter att du har installerat värddatorerna. Verifiera anslutningen mellan värdarna genom att ansluta med SSH från värden i Bastion-säkerhetsgruppen till värdarna i varje undernät.

Exempel: Anslut till en bastion-värd i AWS

1. Konfigurera administratörsdatorn för en ssh-agent. Detta låter dig ansluta till värdar i AWS utan att placera din privata nyckelfil på EC2-instanserna.

   Kör följande kommando för att konfigurera en ssh-agent på en Mac:

   `ssh-add -K myPrivateKey.pem` eller för det senaste operativsystemet på Mac,
   `ssh-add --apple-use-keychain myPrivateKey.pem`

   Se avsnittet Anslut säkert till Linux-instanser som körs i en privat Amazon-VPC för Windows.

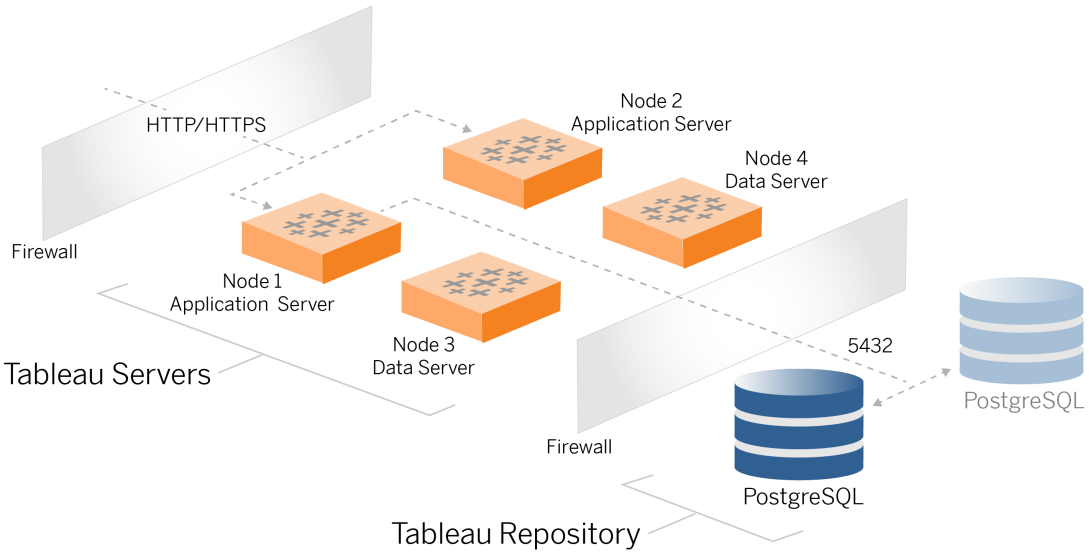
2. Kör följande kommando för att ansluta till bastion-värden:

   ```
   ssh -A ec2-user@<public-IP>
   ```

3. Du kan sedan ansluta till andra värdar i VPC:n från bastion-värden, med den privata IP-adressen, såsom:

```
ssh -A ec2-user@10.0.1.93
```

Del 4 - Installera och konfigurera Tableau Server



Den här ämnet beskriver hur du slutför installation och konfiguration av Tableau Server baslinjedriftsättningen. Proceduren här fortsätter med referensarkitekturexemplet för AWS och Linux.

Exemplen med Linux i installationsprocedurerna visar kommandon för RHEL-liknande distributioner. Mer specifikt har kommandona här utvecklats med Amazon Linux 2-distributionen. Om du kör Ubuntu-driftsättningen redigerar du kommandona på lämpligt sätt.

Innan du börjar

Du måste förbereda och validera din miljö som det beskrivs i Del 3 – förbereda för företagsdriftsättning av Tableau Server.

Installera, konfigurera och tar PostgreSQL

Den här PostgreSQL-instansen är värd för den externa lagringsplatsen för Tableau Server-distributionen. Du måste installera och konfigurera PostgreSQL innan du installerar Tableau.

Du kan köra PostgreSQL på Amazon RDS eller på en EC2-instans. Mer information om skillnaderna mellan att köra lagringsplatsen på en RDS-instans jämfört med en EC2-instans finns i *Extern lagringsplats för Tableau Server* (Linux).

Som ett exempel visar proceduren nedan hur du installerar och konfigurerar Postgres på en Amazon EC2-instans. Exemplet som visas här är en generisk installation och konfiguration för PostgreSQL i referensarkitekturen. Din DBA borde optimera PostgreSQL-driftsättning baserat på storlek på dina data och prestandabehov.

Förhandskrav: observera att du måste köra PostgreSQL 1.6 och du måste installera uuid-ossp-modulen.

PostgreSQL-versionshantering

Du måste installera en kompatibel större version av PostgreSQL för den externa Tableau Server-lagringsplatsen. Dessutom behöver mindre versioner också uppfylla minimikraven.

Tableau Server-versioner	**PostgreSQL-versioner som är kompatibla**
2021.2.3–2021.2.8 2021.3.0–2021.3.7 2021.4.0–2021.4.3	12,6
2021.2.10–2021.2.14 2021.3.8–2021.3.13 2021.4.4–2021.4.8	12,8
2021.2.15–2021.2.16	12.10

2021.3.14–2021.3.15 2021.4.9–2021.4.10	
2021.2.17–2021.2.18 2021.3.16–2021.3.17 2021.4.11–2021.4.12	12,11
2021.3.26 2021.4.23	12,15
2022.1.0	13,3
2022.1.1–2022.1.3	13.4
2022.1.4–2022.1.6	13.6
2022.1.7–2022.1.16 2022.3.0 – 2022.3.7 2023.1.0 – 2023.1.4	13.7
2022.1.17–2022.1.19 2022.3.8 – 2022.3.19 2023.1.5 – 2023.1.15 2023.3.0 – 2023.3.8	13,11
2022.3.20 - 2022.3.x 2023.1.16 - 2023.1.x 2023.3.9 - 2023.3.x	13,14
2024.0–2024.x	15.6

Installera PostgreSQL

I det här exemplet på en installationsprocedur beskrivs hur du installerar PostgreSQL version 13.6.

Logga in på den EC2-värd du skapade i föregående del.

1. Kör en uppdatering för att installera de senaste Linux OS-korrigeringarna:

   ```
   sudo yum update
   ```

2. Skapa och redigera filen pgdg.repo. i sökvägen `/etc/yum.repos.d/`. Lägg till följande konfigurationsinformation i filen:

   ```
   [pgdg13]
   name=PostgreSQL 13 for RHEL/CentOS 7 - x86_64

   baseurl=https://download.postgresql.org/pub/repos/yum/13/redha-
   t/rhel-7-x86_64
   enabled=1
   gpgcheck=0
   ```

3. Så här installerar du Postgres 13.6:

   ```
   sudo yum install postgresql13-server-13.6-1PGDG.rhel7.x86_64
   ```

4. Installera uuid-ossp-modulen:

   ```
   sudo yum install postgresql13-contrib-13.6-1PGDG.rhel7.x86_64
   ```

5. Initiera Postgres:

   ```
   sudo /usr/pgsql-13/bin/postgresql-13-setup initdb
   ```

Konfigurera Postgres

Slutför grundinstallationen genom att konfigurera Postgres:

1. Uppdatera konfigurationsfilen pg_hba, `/var/lib/pgsql/13/data/pg_hba.conf`, med följande två poster. Varje post måste inkludera masken för de undernät där dina Tableau Servers kommer att köras:

   ```
   host all all 10.0.30.0/24 password

   host all all 10.0.31.0/24 password
   ```

2. Uppdatera postgresql-filen, `/var/lib/pgsql/13/data/postgresql.conf`, genom att lägga till följande rad:

   ```
   listen_addresses = '*'
   ```

3. Konfigurera för att starta Postgres vid omstart:

   ```
   sudo systemctl enable --now postgresql-13
   ```

4. Ange ett lösenord för superanvändare:

   ```
   sudo su - postgres

   psql -c "alter user postgres with password 'StrongPassword'"
   ```

   > **Obs**! Ange ett starkt lösenord. Använd inte `'StrongPassword'` som det visas i exemplet här.

   ```
   exit
   ```

5. Starta om Postgres:

   ```
   sudo systemctl restart postgresql-13
   ```

Skapa tar-säkerhetskopia (PostgreSQL steg 1)

Skapa en tar-säkerhetskopia av PostgreSQL-konfigurationen. Genom att skapa en tar-ögonblicksbild av den aktuella konfigurationen kan du spara tid om du får problem när du

fortsätter med driftsättningen.

Vi kallar detta för ”Steg 1”-säkerhetskopian.

På PostgreSQL-värden:

1. Stoppa Postgres-databasinstansen:

   ```
   sudo systemctl stop postgresql-13
   ```

2. Skapa säkerhetskopian genom att köra följande kommandon:

   ```
   sudo su
   cd /var/lib/pgsql
   tar -cvf step1.13.bkp.tar 13
   exit
   ```

3. Starta Postgres-databasen:

   ```
   sudo systemctl start postgresql-13
   ```

Återställa till steg 1

Återställ till Steg 1 om den ursprungliga Tableau Server-noden upplever ett fel under installationen.

1. På datorn som kör Tableau kör du skriptet obliterate för att fullständigt ta bort Tableau Server från värden:

   ```
   sudo /app/tableau_server/packages/scripts.<version_
   code>/./tableau-server-obliterate -a -y -y -y -l
   ```

2. Återställ PostgreSQL steg 1 tar. Kör följande kommandon på datorn där Postgres körs:

   ```
   sudo su
   systemctl stop postgresql-13
   ```

```
cd /var/lib/pgsql

tar  -xvf step1.13.bkp.tar

systemctl start postgresql-13

exit
```

Återuppta installationsprocessen med att installera den ursprungliga noden av Tableau Server.

Innan du installerar

Om du driftsätter Tableau enligt exempelimplementeringen för AWS/Linux som beskrivs i den här guiden så är det möjligt att du kan köra det automatiska installationsskriptet TabDeploy4EDG. TabDeploy4EDG-skriptet automatiserar exempelinstallationen av Tableau-driftsättningen med fyra noder som beskrivs i procedurerna som följer. Se Bilaga – AWS Deployment Toolbox.

Installera den ursprungliga noden för Tableau Server

Den här processen beskriver hur du installerar den ursprungliga noden av Tableau Server som det definieras av referensarkitekturen. Med undantag för paketinstallationen och initieringen av TSM använder sig den här proceduren av TSM-kommandoraden närhelst det är möjligt. Utöver att vara plattformsagnostisk så tillåter TSM CLI en mer sömlös installation i virtualiserade och huvudlösa miljöer.

Kör installationspaketet och initiera TSM

Logga in på värdservern till nod 1.

1. Kör en uppdatering för att installera de senaste Linux OS-korrigeringarna:

   ```
   sudo yum update
   ```

2. Kopiera installationspaketet från sidan Tableau-nedladdningar till den värddator som kommer att köra Tableau Server.

   På en dator som kör ett Linux RHEL-liknande operativsystem kör du till exempel:

   ```
   wget
   https://downloads.tableau.com/esdalt/2022<version>/tableau-
   server-<version>.rpm
   ```

   där `<version>` är versionsnumret.

3. Ladda ner och installera beroenden:

   ```
   sudo yum deplist tableau-server-<version>.rpm | awk
   '/provider:/ {print $2}' | sort -u | xargs sudo yum -y install
   ```

4. Skapa sökvägen `/app/tableau_server` i rotkatalogen:

   ```
   sudo mkdir -p /app/tableau_server
   ```

5. Kör installationsprogrammet och ange installationssökvägen `/app/tableau_server`. På ett Linux RHEL-liknande operativsystem kör du till exempel:

   ```
   sudo rpm -i --prefix /app/tableau_server tableau-server-
   <version>.x86_64.rpm
   ```

6. Ändra till katalogen `/app/tableau_server/packages/scripts.<version_code>/` och kör skriptet `initialize-tsm` som du hittar där:

   ```
   sudo ./initialize-tsm -d /data/tableau_data --accepteula
   ```

7. Efter att initieringen slutförts avslutar du terminalsessionen:

   ```
   exit
   ```

Aktivera och registrera Tableau Server

1. Logga in på värdservern till nod 1.
2. Ange Tableau Server-produktnycklarna i detta steg. Kör följande kommando för varje licensnyckel som du har köpt:

   ```
   tsm licenses activate -k <product key>
   ```

3. Skapa en JSON-registreringsfil med formatet som visas här:

   ```
   {
   "zip" : "97403",
   "country" : "USA",
   "city" : "Springfield",
   "last_name" : "Simpson",
   "industry" : "Energy",
   "eula" : "yes",
   "title" : "Safety Inspection Engineer",
   "company_employees" : "100",
   "phone" : "5558675309",
   "company" : "Example",
   "state" : "OR",
   "opt_in" : "true",
   "department" : "Engineering",
   "first_name" : "Homer",
   "email" : "homer@example.com"
   }
   ```

4. När du har sparat ändringarna i filen skickar du den tillsammans med alternativet `--file` för att registrera Tableau Server:

   ```
   tsm register --file path_to_registration_file.json
   ```

Konfigurera identitetslagret

Obs! Om din driftsättning kommer att använda sig av ett externt lager för Tableau-fillagret så behöver du aktivera Extern fillagring innan du konfigurerar identitetslagret. Läs mer i *Installera Tableau Server med extern fillagring* (Linux).

Standardreferensarkitekturen använder sig av ett lokalt identitetslager. Konfigurera den ursprungliga värden med lokalt identitetslager genom att skicka `config.json`-filen med kommandot `tsm settings import`.

Importera filen `config.json` i enlighet med ditt operativsystem:

Filen `config.json` ingår i katalogsökvägen för scripts.<version> (till exempel, `scripts.20204.21.0217.1203`), och formateras för att konfigurera identitetslagret.

Kör följande kommando för att importera filen `config.json`:

```
tsm settings import -f /app/tableau_
server/packages/scripts.<version_code>/config.json
```

Konfigurera extern Postgres

1. Skapa en extern databas JSON-fil med följande konfigurationsinställningar:

   ```
   {
   "flavor":"generic",
   "masterUsername":"postgres",
   "host":"<instance ip address>",
   "port":5432
   }
   ```

2. När du sparat ändringarna till filen skickar du den med följande kommando:

```
tsm topology external-services repository enable -f
<filename>.json --no-ssl
```

Du uppmanas att ange lösenordet för Postgres-huvudanvändaren.

Alternativet `--no-ssl` konfigurerar Tableau att endast använda SSL/TLS när Postgres-servern är konfigurerad för SSL/TLS. Om Postgres inte är konfigurerat för SSL/TLS är anslutningen inte krypterad. Del 6 - Konfigurera efter installation beskriver hur du aktiverar SSL/TLS för Postgres-anslutningen efter att du har slutfört den första fasen av driftsättningen.

3. Tillämpa ändringarna.

   Kör det här kommandot för att tillämpa ändringarna och starta om Tableau Server:

   ```
   tsm pending-changes apply
   ```

4. Ta bort den konfigurationsfil du använde i Steg 1.

Avsluta installationen av nod 1

1. Efter att Tableau Server installerats så måste du initiera servern.

   Kör följande kommando:

   ```
   tsm initialize --start-server --request-timeout 1800
   ```

2. När initieringen avslutats så måste du skapa ett administratörskonto för Tableau Server.

   Till skillnad från det datakonto du använder för att installera och hantera komponenterna för TSM-operativsystemet så är administratörskontot för Tableau Server ett programkonto som används för att skapa Tableau Server-användare, -projekt och -platser. Tableau Server-administratören applicerar även behörigheter på Tableau-resurser. Skapa det initiala administratörskontot genom att köra följande kommando: I följande exempel heter användaren `tableau-admin`:

```
tabcmd initialuser --server http://localhost --
username "tableau-admin"
```

Tabcmd uppmanar dig att ange ett lösenord för den här användaren.

Verifiering: konfiguration av nod 1

1. Verifiera att TSM-tjänsterna körs genom att köra följande kommando:

   ```
   tsm status -v
   ```

   Tableau borde returnera följande:

   ```
   external:
   Status: RUNNING
   'Tableau Server Repository 0' is running (Active Repository).
   node1: localhost
   Status: RUNNING
   'Tableau Server Gateway 0' is running.
   'Tableau Server Application Server 0' is running.
   'Tableau Server Interactive Microservice Container 0' is
   running.
   'MessageBus Microservice 0' is running.
   'Relationship Query Microservice 0' is running.
   'Tableau Server VizQL Server 0' is running.
   ...
   ```

   Alla tjänster listas.

2. Verifiera att den Tableau-administrativa platsen körs genom att köra följande kommando:

   ```
   curl localhost
   ```

   De första raderna borde visa Vizportal html som liknar detta:

   ```
   <!DOCTYPE html>
   <html xmlns:ng="" xmlns:tb="">
   <head ng-csp>
   ```

```
<meta charset="UTF-8">
<meta http-equiv="X-UA-Compatible" content="IE=edge">
<meta name="viewport" content="initial-scale=1, maximum-
scale=2, width=device-width, height=device-height, viewport-
fit=cover">
<meta name="format-detection" content="telephone=no">
<meta name="vizportal-config ...
```

Skapa tar-säkerhetskopior (steg 2)

Skapa två tar-säkerhetskopior efter att du verifierat den inledande installationen:

- PostgreSQL
- Ursprunglig Tableau-nod (nod 1)

I de flesta fall kan du återställa din installation av den ursprungliga noden genom att återställa dessa tar-filer. Det går mycket snabbare att återställa tar-filerna än att ominstallera och återinitiera den ursprungliga noden.

Skapa tar-säkerhetskopior (steg 2)

1. Stoppa Tableau på den ursprungliga Tableau-noden:

   ```
   tsm stop
   ```

   Vänta tills Tableau stoppar innan du fortsätter till nästa steg.

2. Stoppa Postgres-databasinstansen på PostgreSQL-värden:

   ```
   sudo systemctl stop postgresql-13
   ```

3. Skapa säkerhetskopian genom att köra följande kommandon:

   ```
   sudo su
   ```

   ```
   cd /var/lib/pgsql
   ```

```
tar -cvf step2.13.bkp.tar 13

exit
```

4. Verifiera att Postgres tar-filen skapats med rotbehörigheter:

   ```
   sudo ls -al /var/lib/pgsql
   ```

5. På Tableau-värden stoppar du Tableau-administrativa tjänster:

   ```
   sudo /app/tableau_server/packages/scripts.<version_
   code>/./stop-administrative-services
   ```

6. Skapa säkerhetskopian genom att köra följande kommandon:

   ```
   cd /data

   sudo tar -cvf step2.tableau_data.bkp.tar tableau_data
   ```

7. Starta Postgre-databasen på Postgres-värden:

   ```
   sudo systemctl start postgresql-13
   ```

8. Starta Tableau-administrativa tjänster:

   ```
   sudo /app/tableau_server/packages/scripts.<version_
   code>/./start-administrative-services
   ```

9. Kör kommandot `tsm status` för att övervaka statusen för TSM innan du startar om.

   I de flesta fall returnerar kommandot först status DEGRADED eller ERROR. Vänta några minuter och kör kommandot igen. Om statusen ERROR eller DEGRADED returneras fortsätter du att vänta. Försök inte starta TSM förrän statusen STOPPED returneras. Kör sedan följande kommando:

   ```
   tsm start
   ```

Återställa till steg 2

Den här processen återställer nod 1 i Tableau och Postgres-instansen till steg 2. Efter att du återställt till det här steget så kan du omdistribuera de kvarvarande Tableau-noderna.

1. Stoppa TSM-tjänsterna på den ursprungliga Tableau-värden (nod 1):

   ```
   tsm stop
   ```

2. Stoppa Tableau-administrativa tjänster på alla noder för Tableau Server-distributionen. Kör följande kommando på noderna i rätt ordning (nod 1, nod 2 och nod 3):

   ```
   sudo /app/tableau_server/packages/scripts.<version_
   code>/./stop-administrative-services
   ```

3. Efter att Tableau-tjänsterna stoppats återställer du tar-filen för PostgreSQL steg 2. Kör följande kommandon på datorn där Postgres körs:

   - ```
     sudo su

     systemctl stop postgresql-13

     cd /var/lib/pgsql

     tar  -xvf step2.13.bkp.tar

     systemctl start postgresql-13

     exit
     ```

4. Återställ tar-fil för PostgreSQL steg 2. Kör följande kommandon på den ursprungliga Tableau-värden:

   ```
   cd /data

   sudo rm -rf tableau_data

   sudo tar  -xvf step2.tableau_data.bkp.tar
   ```

5. Ta bort följande filer på den första Tableau-noden (nod 1):

   - ```
     sudo rm /data/tableau_
     data/data/tabsvc/appzookeeper/0/version-2/currentEpoch
     ```

- `sudo rm /data/tableau_data/data/tabsvc/appzookeeper/0/version-2/acceptedEpoch`
- `sudo rm /data/tableau_data/data/tabsvc/tabadminagent/0/servicestate.json`

6. Starta Tableau-administrativa tjänster:

   ```
   sudo /app/tableau_server/packages/scripts.<version_code>/./start-administrative-services
   ```

7. Ladda om Tableau systmctl-filerna och kör sedan `start-administrative-services` igen:

   ```
   sudo su -l tableau -c "systemctl --user daemon-reload"
   ```

   ```
   sudo /app/tableau_server/packages/scripts.<version_code>/./start-administrative-services
   ```

8. Kör kommandot `tsm status` på nod 1 för att övervaka statusen för TSM på nod 1 innan du startar om.

   Ibland kan du få ett fel av typen `Cannot connect to server...`. Det här felet beror på att tabadmincontroller-tjänsten inte har startats om. Fortsätt att köra `tsm status` regelbundet. Om felet kvarstår efter tio minuter kör du kommandot `start-administrative-services` igen.

   Efter en stund returnerar kommandot `tsm status` statusen DEGRADED och sedan ERROR. Starta inte TSM förrän statusen STOPPED returneras. Kör sedan följande kommando:

   ```
   tsm start
   ```

Återuppta installationsprocessen för att installera Tableau Server på de kvarvarande noderna.

Installera Tableau Server på kvarvarande noder

Fortsätt distributionen genom att kopiera Tableau-installationsprogrammet till varje nod.

Översikt över nodkonfigurationen

Det här avsnittet beskriver processen för att konfigurera nod 2–4. De avsnitt som följer ger dig detaljerade procedurer för konfiguration och validering av varje steg.

Installation av Tableau Server-noderna 2–4 kräver att du genererar, kopierar och refererar till en startfil under nodinstallationen.

Generera startfilen genom att köra ett TSM-kommando på den ursprungliga noden. Du kopierar därefter startfilen till målnoden där du kör den som en del av nodinitieringen.

Följande JSON-innehåll visar på ett exempel på en startfil. (Certifikatet och kryptorelaterade värden har trunkerats för att göra exempelfilen enklare att läsa.)

```
{
"initialBootstrapSettings" : {
       "certificate" : "-----BEGIN CERTIFICATE-----\r\...=\r\n-----END
CERTIFICATE-----",
       "port" : 8850,
       "configurationName" : "tabsvc",
       "clusterId" : "tabsvc-clusterid",
       "cryptoKeyStore" : "zs7OzgAAAAIAAAABAAAAA...w==",
       "toksCryptoKeystore" : "LS0tLS1CRUdJTiBUT00tLS0tCjM5MDBh...L",
       "sessionCookieMaxAge" : 7200,
       "nodeId" : "node1",
       "machineAddress" : "ip-10-0-1-93.us-west-1.compute.internal",
       "cryptoEnabled" : true,
       "sessionCookieUser" : "tsm-bootstrap-user",
       "sessionCookieValue" :
"eyJjdHkiOiJKV1QiLCJlbmMiOiJBMTI4Q0JDLUhQ...",
```

```
        "sessionCookieName" : "AUTH_COOKIE"
        }
}
```

Startfilen inkluderar anslutningsbaserad validering för att autentisera nod 1 och skapar en krypterad kanal för startfilsprocessen. Startfilssessionen är tidsbegränsad och konfiguration samt validering av noder är tidskrävande. Planera att skapa och kopiera nya startfiler allteftersom du konfigurerar noderna.

Efter att du kör startfilen så loggar du in på den ursprungliga Tableau Server-noden och konfigurerar processerna för den nya noden. När du är klar med att konfigurera noderna så måste du tillämpa ändringarna och starta om den ursprungliga noden. Den nya noden är konfigurerad och startad. Allteftersom du lägger till noder kommer konfiguration och omstart av distributionen att ta allt längre att slutföra.

Exemplen med Linux i installationsprocedurerna visar kommandon för RHEL-liknande distributioner. Om du kör Ubuntu-driftsättningen redigerar du kommandona på lämpligt sätt.

1. Kör en uppdatering för att installera de senaste Linux OS-korrigeringarna:

   ```
   sudo yum update
   ```

2. Ladda ner och installera beroenden:

   ```
   sudo yum deplist tableau-server-<version>.rpm | awk
   '/provider:/ {print $2}' | sort -u | xargs sudo yum -y install
   ```

3. Skapa sökvägen `/app/tableau_server` i rotkatalogen:

   ```
   sudo mkdir -p /app/tableau_server
   ```

4. Kör installationsprogrammet och ange installationssökvägen `/app/tableau_server`. På ett Linux RHEL-liknande operativsystem kör du till exempel:

   ```
   sudo rpm -i --prefix /app/tableau_server tableau-server-
   <version>.x86_64.rpm
   ```

Generera, kopiera och använda startfilen för att initiera TSM

Följande procedur visar hur du genererar, kopierar och använder en startfil när du initierar TSM på en annan nod. I det här exemplet heter startfilen `boot.json`.

I det här exemplet kör värddatorerna AWS och EC2-värdarna kör Amazon Linux 2.

1. Anslut till den ursprungliga noden (nod 1) och kör följande kommando:

   ```
   tsm topology nodes get-bootstrap-file --file boot.json
   ```

2. Kopiera startfilen till nod 2.

   ```
   scp boot.json ec2-user@10.0.31.83:/home/ec2-user/
   ```

3. Anslut till nod 2 och växla till Tableau Server-skriptkatalogen:

   ```
   cd /app/tableau_server/packages/scripts.<version_number>
   ```

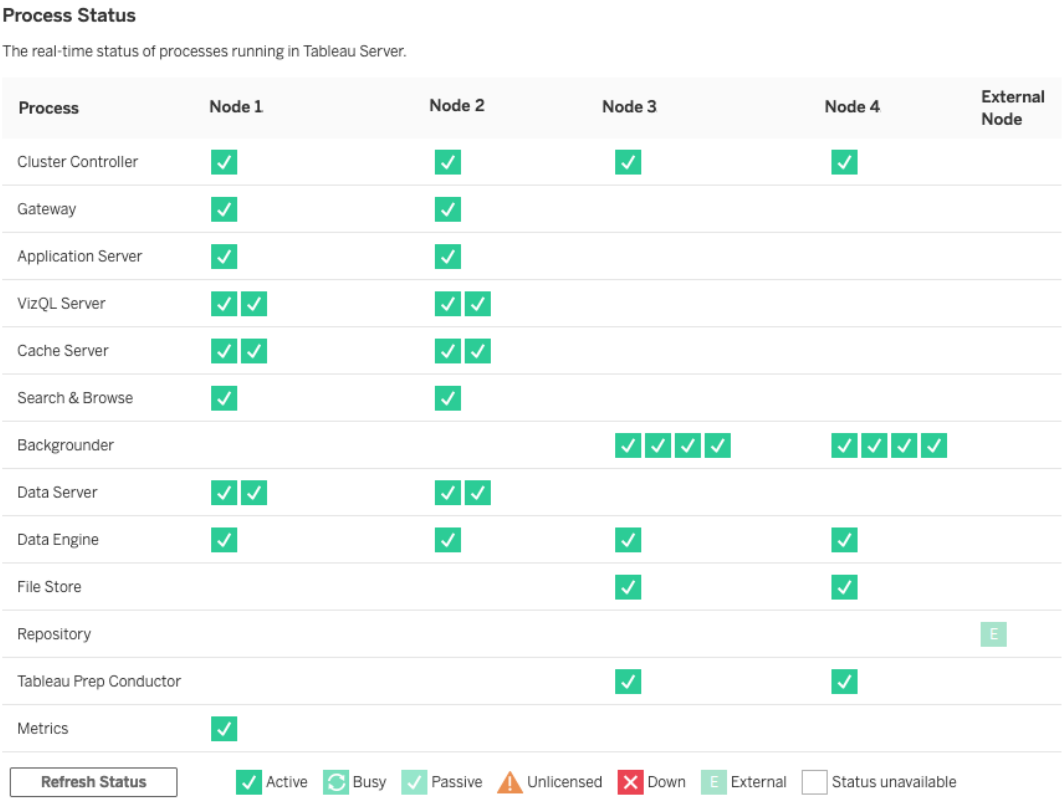
4. Kör kommandot `initialize-tsm` och referera till startfilen:

   ```
   sudo ./initialize-tsm -d /data/tableau_data -b /home/ec2-
   user/boot.json --accepteula
   ```

5. Efter `initialize-tsm` har slutförts tar du bort `boot.json` och avslutar eller loggar ut från sessionen.

Konfigurera processer

Du måste konfigurera Tableau Server-klustret på noden där Tableau Server Administration Controller (TSM controller) körs. TSM-styrenheten körs på den ursprungliga noden.



Konfigurera nod 2

1. Efter att du initierat TSM med hjälp av startfilen på nod 2 så loggar du in på den ursprungliga noden.

2. På den ursprungliga noden (`node1`) kör du följande kommandon för att konfigurera processer på nod 2:

```
tsm topology set-process -n node2 -pr clustercontroller -c 1
tsm topology set-process -n node2 -pr gateway -c 1
tsm topology set-process -n node2 -pr vizportal -c 1
tsm topology set-process -n node2 -pr vizqlserver -c 2
tsm topology set-process -n node2 -pr cacheserver -c 2
tsm topology set-process -n node2 -pr searchserver -c 1
tsm topology set-process -n node2 -pr dataserver -c 2
```

```
tsm topology set-process -n node2 -pr clientfileservice -c 1
tsm topology set-process -n node2 -pr tdsservice -c 1
tsm topology set-process -n node2 -pr collections -c 1
tsm topology set-process -n node2 -pr contentexploration -c 1
```

Om du installerar version 2022.1 eller senare ska du också lägga till tjänsten Index och Sök:

```
tsm topology set-process -n node2 -pr indexandsearchserver -c 1
```

Om du installerar version 2023.3 eller senare inkluderar du bara index- och sökservern. Lägg inte till tjänsten Sök och bläddra (searchserver).

3. Granska konfigurationen innan du tillämpar den. Kör följande kommando:

   ```
   tsm pending-changes list
   ```

4. Efter att du verifierat att dina ändringar är i listan med väntande (det kommer även att finnas andra tjänster i listan med väntande) så tillämpar du ändringarna:

   ```
   tsm pending-changes apply
   ```

   En omstart krävs för att ändringarna ska tillämpas. Konfigurationen och omstarten kan ta en stund.

5. Verifiera konfigurationen för nod 2. Kör följande kommando:

   ```
   tsm status -v
   ```

Konfigurera nod 3

Initiera TSM med hjälp av startfilsprocessen på nod 3 och kör sedan kommandona `tsm topology set-process` nedan.

Det finns en samordningstjänstvarning som visas varje gång du anger en process. Du kan ignorera varningen när du anger processerna.

1. Efter att du initierar TSM med hjälp av startfilen på nod 3 loggar du in på den ursprungliga noden (`node1`) och kör följande kommandon för att konfigurera processer:

   ```
   tsm topology set-process -n node3 -pr clustercontroller -c 1
   tsm topology set-process -n node3 -pr clientfileservice -c 1
   tsm topology set-process -n node3 -pr backgrounder -c 4
   tsm topology set-process -n node3 -pr filestore -c 1
   ```

   Om du installerar version 2022.1 eller senare ska du också lägga till tjänsten Index och Sök:

   ```
   tsm topology set-process -n node3 -pr indexandsearchserver -c 1
   ```

2. Granska konfigurationen innan du tillämpar den. Kör följande kommando:

   ```
   tsm pending-changes list
   ```

3. Efter att du verifierat att dina ändringar är i listan med väntande (listan kommer att inkludera andra tjänster som konfigureras automatiskt) så tillämpar du ändringarna:

   ```
   tsm pending-changes apply --ignore-warnings
   ```

   En omstart krävs för att ändringarna ska tillämpas. Konfigurationen och omstarten kan ta en stund.

4. Verifiera konfigurationen genom att köra följande kommando:

   ```
   tsm status -v
   ```

Driftsätt samordningstjänstensemblen till nod 1-3

För fyranodsdistributioner med standardreferensarkitektur kör du följande procedur:

1. Kör följande kommandon på nod 1:

   ```
   tsm stop
   tsm topology deploy-coordination-service -n node1,node2,node3
   ```

   Processen inkluderar en omstart av TSM vilket tar lite tid.

2. Efter att samordningstjänsten har driftsatts startar du TSM:

   ```
   tsm start
   ```

Skapa tar-säkerhetskopior (steg 3)

Skapa fyra tar-säkerhetskopior efter att du verifierat installationen:

- PostgreSQL
- Ursprunglig Tableau-nod (nod 1)
- Nod 2 i Tableau
- Nod 3 i Tableau

Skapa tar-säkerhetskopior (steg 3)

1. Stoppa Tableau på den ursprungliga Tableau-noden:

   ```
   tsm stop
   ```

2. Efter att TSM har stoppat stoppar du Tableau-administrativa tjänster på varje nod. Kör följande kommando på noderna i rätt ordning (nod 1, nod 2 och nod 3):

   ```
   sudo /app/tableau_server/packages/scripts.<version_
   code>/./stop-administrative-services
   ```

3. Stoppa Postgres-databasinstansen på PostgreSQL-värden:

   ```
   sudo systemctl stop postgresql-12
   ```

4. Skapa säkerhetskopian genom att köra följande kommandon:

   ```
   sudo su

   cd /var/lib/pgsql

   tar -cvf step3.12.bkp.tar 12

   exit
   ```

5. Verifiera att Postgres tar-filen skapats med rotbehörigheter:

   ```
   sudo ls -al /var/lib/pgsql
   ```

6. Starta Postgre-databasen på Postgres-värden:

   ```
   sudo systemctl start postgresql-12
   ```

7. Skapa tar-säkerhetskopian på nod 1, nod 2 och nod 3. Kör följande kommando på alla noder:

   - `cd /data`

     ```
     sudo tar -cvf step3.tableau_data.bkp.tar tableau_data
     ```

   - Verifiera att Tableau tar-filen skapats med rotbehörigheter:

     ```
     ls -al
     ```

8. Starta Tableau-administrativa tjänster på varje nod i ordningen (nod 1, nod 2 och nod 3):

   ```
   sudo /app/tableau_server/packages/scripts.<version_
   code>/./start-administrative-services
   ```

9. Kör kommandot `tsm status` för att övervaka statusen för TSM innan du startar om.

   I de flesta fallen returnerar kommandot status DEGRADED och därefter ERROR. Vänta några minuter och kör kommandot igen. Om statusen ERROR eller

DEGRADED returneras fortsätter du att vänta. Försök inte starta TSM förrän statusen STOPPED returneras. Kör sedan följande kommando:

```
tsm start
```

Återställa till steg 3

Den här processen återställer Tableau nod 1, nod 2 och nod 3. Den återställer även Postgres-instansen till steg 3. Efter att du återställt till det här steget kan du driftsätta samordningstjänsten, nod 4 och därefter slutliga nodkonfigurationer.

1. Stoppa TSM-tjänsten på den ursprungliga Tableau-värden (nod 1):

   ```
   tsm stop
   ```

2. Efter att TSM har stoppat, stoppar du Tableau-administrativa tjänster på nod 1, nod 2 och nod 3. Kör följande kommando på varje nod:

   ```
   sudo /app/tableau_server/packages/scripts.<version_
   code>/./stop-administrative-services
   ```

3. Återställ tar-filen för PostgreSQL steg 3. Kör följande kommandon på datorn där Postgres körs:

   ```
   sudo su
   systemctl stop postgresql-12
   cd /var/lib/pgsql
   tar -xvf step3.12.bkp.tar
   systemctl start postgresql-12
   exit
   ```

4. Återställ tar-filen för Tableau steg 3 på nod 1, nod 2 och nod 3. Kör följande kommandon på alla Tableau-noder:

   ```
   cd /data

   sudo rm -rf tableau_data

   sudo tar  -xvf step3.tableau_data.bkp.tar
   ```

5. Ta bort följande filer på den första Tableau-noden (nod 1):

   - `sudo rm /data/tableau_data/data/tabsvc/appzookeeper/1/version-2/currentEpoch`
   - `sudo rm /data/tableau_data/data/tabsvc/appzookeeper/1/version-2/acceptedEpoch`
   - `sudo rm /data/tableau_data/data/tabsvc/tabadminagent/0/servicestate.json`

   Om kommandotolken returnerar felet ”filen hittades inte” så kan du behöva ändra sökvägsnamnet för att öka numret `<n>` i den här delen av sökvägen: `.../appzookeeper/<n>/version-2/....`

6. Starta om administrativa tjänster på nod 1, nod 2 och nod 3. Kör följande kommando på alla noder:

   ```
   sudo /app/tableau_server/packages/scripts.<version_
   code>/./start-administrative-services

   sudo su -l tableau -c "systemctl --user daemon-reload"

   sudo /app/tableau_server/packages/scripts.<version_
   code>/./start-administrative-services
   ```

7. Kör kommandot `tsm status` på nod 1 för att övervaka statusen för TSM på nod 1 innan du startar om.

Ibland kan du få ett fel av typen `Cannot connect to server...`. Det här felet beror på att tabadmincontroller-tjänsten inte har startats om. Fortsätt att köra `tsm status` regelbundet. Om felet kvarstår efter tio minuter kör du kommandot `start-administrative-services` igen.

Efter en stund returnerar kommandot `tsm status` statusen DEGRADED och sedan ERROR. Starta inte TSM förrän status STOPPED returneras. Kör sedan följande kommando:

```
tsm start
```

Återupptar installationsprocessen för att driftsätta samordningstjänsten på nod 1–3.

Konfigurera nod 4

Processen för att konfigurera nod 4 är densamma som för nod 3.

Ange samma processer som du anger för nod 3 och kör samma kommandon som visas ovan men ange `node4` i kommandona i stället för `node3`.

Precis som vid nod 3-verifiering verifierar du nod 4-konfigurationen genom att köra `tsm status -v`.

Vänta tills fillagringsprocessen på Nod 4 har slutfört synkroniseringen innan du går vidare. Fillagringsstatusen visar `is synchronizing` tills den har slutförts. När fillagringsstatusen visar `is running` kan du gå vidare.

Slutlig processkonfiguration och verifiering

Det slutliga steget för att bearbeta konfigurationen är att ta bort överflödiga processer från nod 1.

1. Anslut till den ursprungliga noden (`node1`).
2. Ta fillagret på nod 1 ur drift. Det här ger en varning om att ta bort fillagret från en samlokaliserad styrenhet. Du kan ignorera varningen. Kör följande kommando:

   ```
   tsm topology filestore decommission -n node1
   ```

3. När fillagret har tagits ur drift kör du följande kommando för att ta bort bakgrundsprocessen från nod 1:

   ```
   tsm topology set-process -n node1 -pr backgrounder -c 0
   ```

4. Granska konfigurationen innan du tillämpar den. Kör följande kommando:

   ```
   tsm pending-changes list
   ```

5. Efter att du verifierat att dina ändringar är i listan med väntande så tillämpar du ändringarna:

   ```
   tsm pending-changes apply
   ```

   En omstart krävs för att ändringarna ska tillämpas. Konfigurationen och omstarten kan ta en stund.

6. Verifiera konfigurationen:

   ```
   tsm status -v.
   ```

   Vänta tills fillagringsprocessen på Nod 4 har slutfört synkroniseringen innan du går vidare. Fillagringsstatusen visar `is synchronizing` tills den har slutförts. När fillagringsstatusen visar `is running` kan du gå vidare.

Utför säkerhetskopiering

En fullständig återställning av Tableau Server kräver en säkerhetskopieringsportfölj som inkluderar tre komponenter:

- En säkerhetskopia av lagringsplatsen och fillagerdata. Den här filen genereras av kommandot `tsm maintenance backup`.
- En exportfil med topologi och konfiguration. Den här filen genereras av kommandot `tsm settings export`.
- Autentiseringscertifikat-, nyckel- och keytabfiler.

En fullständig beskrivning av säkerhetskopierings- och återställningsprocessen finns i Tableau Server-ämnet *Utför en fullständig säkerhetskopiering och återställning av Tableau Server* (Linux).

Vid det här stadiet i din driftsättning inkluderas alla relevanta filer och tillgångar som krävs för en fullständig återställning genom att köra kommandona `tsm maintenance backup` och `tsm settings export`.

1. Kör följande kommando för att exportera konfigurationen och topologiinställningarna till en fil som kallas `ts_settings_backup.json`

   ```
   tsm settings export -f ts_settings_backup.json
   ```

2. Kör följande kommando för att skapa en säkerhetskopia av lagringsplatsen och fillagerdata i en fil som kallas `ts_backup-<yyyy-mm-dd>.tsbak`. Ignorera varningen om att fillagret inte finns på styrenhetens nod.

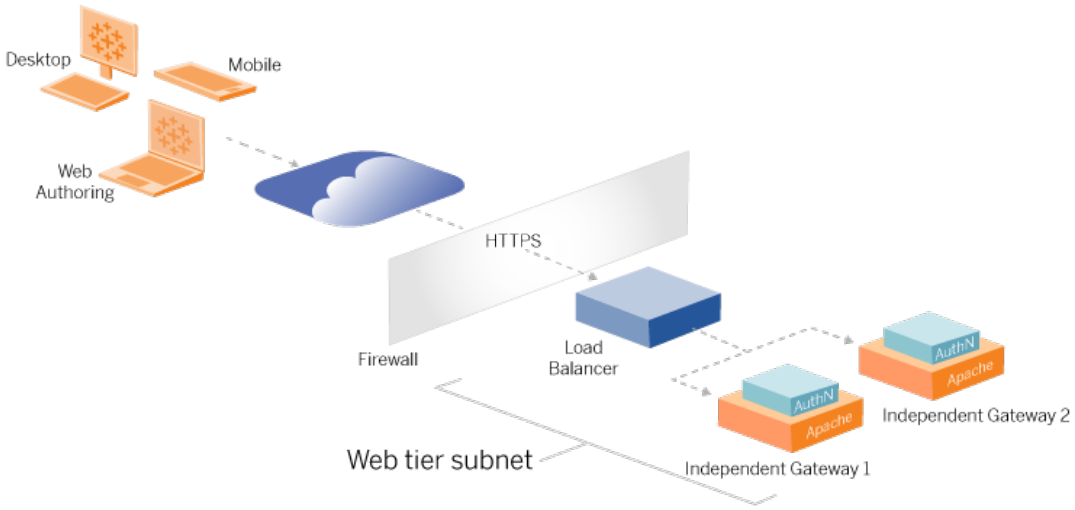
   ```
   tsm maintenance backup -f ts_backup -d --skip-compression
   ```

   Plats för säkerhetskopian:

   ```
   /data/tableau_data/data/tabsvc/files/backups/
   ```

3. Kopiera bägge filerna och spara dem på en annan lagringstillgång som inte delas av din Tableau Server-driftsättning.

Del 5 - Konfigurera webbnivån



Webbnivån för referensarkitekturen ska inkludera följande komponenter:

- En belastningsutjämnare för webborienterade program som accepterar HTTPS-begäranden från Tableau-klienter och som kommunicerar med de omvända proxyservrarna.
- Omvänd proxy:
  - Vi rekommenderar att du driftsätter den oberoende gatewayen för Tableau Server.
  - Vi rekommenderar minst två proxyservrar för redundans och för att hantera klientbelastningen.
  - Tar emot HTTPS-trafik från belastningsutjämnaren.
  - Stöder ihållande sessioner mot Tableau-värden.
  - Konfigurera proxy för belastningsutjämning med resursallokering (round robin) för varje Tableau Server som kör gatewayprocessen.
  - Hanterar autentiseringsbegäranden från en extern identitetsprovider (IdP).
- Proxy för vidarebefordran: Tableau Server kräver åtkomst till Internet för licensierings- och kartfunktioner. Du måste konfigurera listor över säkra proxyer för vidarebefordran för Tableau-tjänste-URL:er. Läs mer i *Kommunicera med Internet* (Linux).

- All klientrelaterad trafik kan krypteras över HTTPS:
  - Lastbalanserare för klient till program
  - Lastbalanserare för omvända proxyservrar
  - Proxyservrar till Tableau Server
  - Autentiseringshanterare som körs på omvänd proxy till IdP
  - Tableau Server till IdP

Oberoende gateway för Tableau Server

Den oberoende gatewayen för Tableau Server introducerades i Tableau Server version 2022.1. Oberoende gateway är en fristående instans av gatewayprocessen i Tableau som fungerar som en Tableau-medveten omvänd proxy.

Oberoende gateway har stöd för belastningsutjämning med resursallokering (round robin) till Tableaus backend-servrar. Oberoende gateway är emellertid inte avsedd att fungera som belastningsutjämnare för företagsprogram. Du bör köra oberoende gateway bakom en belastningsutjämnare i företagsklass.

Den här funktionen kräver en licens för Advanced Management.

Autentisering och auktorisering

Standardreferensarkitekturen specificerar att installera Tableau Server med lokal autentisering konfigurerad. I den här modellen måste klienter ansluta till Tableau Server för att autentiseras av den Tableau Server-interna lokala autentiseringsprocessen. Vi rekommenderar inte att du använder den här autentiseringsmetoden i referensarkitekturen eftersom scenariot kräver att oautentiserade klienter kommunicerar på programnivån vilket utgör en säkerhetsrisk.

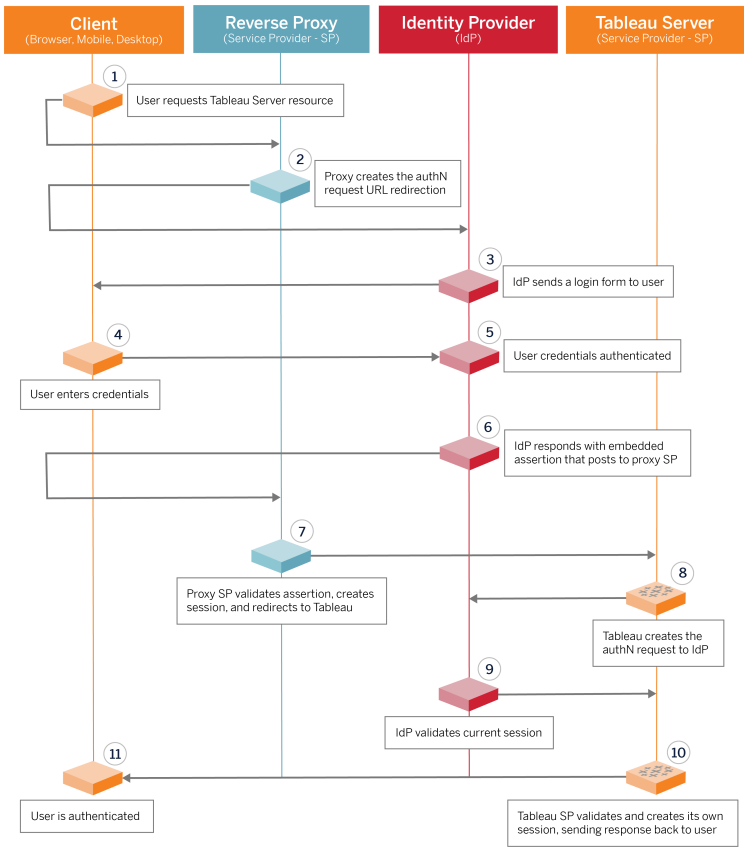
I stället rekommenderar vi att du konfigurerar en extern identitetsprovider i företagsklass kopplad till en AuthN-modul för att förautentisera all trafik till programnivån. När den konfigurerats med en extern IdP används inte den Tableau Server-interna lokala autentiseringsprocessen. Tableau Server godkänner åtkomst till resurser i distributionen efter att IdP har autentiserat användarna.

Förautentisering med en AuthN-modul

I det exempel som dokumenteras i den här guiden konfigureras SAML SSO, men förautentiseringsprocessen kan konfigureras med de flesta externa identitetsprovidrar och en AuthN-modul.

I referensarkitekturen konfigureras den omvända proxyn för att skapa en klientautentiseringssession med IdP innan de begärandena proxyas till Tableau Server. Vi kallar den här processen för *förautentiserings*fasen. Den omvända proxyn omdirigerar endast autentiserade klientsessioner till Tableau Server. Tableau Server skapar därefter en session, verifierar autentiseringen av sessionen med IdP och returnerar klientbegäran.

Följande diagram visar de stegvisa detaljerna för förautentiserings- och autentiseringsprocessen med en AuthN-modul konfigurerad. Den omvända proxyn kan vara en generisk tredjepartslösning eller en oberoende gateway för Tableau Server:



Konfigurationsöversikt

Det här är en översikt för processen att konfigurera webbnivån. Verifiera anslutning efter varje steg:

1. Konfigurera två omvända proxyer för att ge HTTP-åtkomst till Tableau Server
2. Konfigurera logiken för belastningsutjämning med ihållande sessioner att ansluta till varje Tableau Server-instans som kör Gateway-processen.
3. Konfigurera belastningsutjämning för program med ihållande sessioner på internet-gatewayen som vidarebefordrar begäranden till de omvända proxyservrarna.
4. Konfigurera autentisering med en extern IdP. Du kan konfigurera SSO eller SAML genom att installera en autentiseringshanterare på de omvända proxyservrarna. AuthN-modulen hanterar autentiseringshandskakningen mellan den externa IdP och din Tableau-driftsättning. Tableau agerar även som en IdP-tjänstprovider och autentiserar användare med IdP.
5. Om du vill autentisera med Tableau Desktop i den här distributionen så måste dina klienter köra Tableau Desktop 2021.2.1 eller senare.

Exempel på webbnivåkonfiguration med oberoende gateway för Tableau Server

Den resterande delen av det här ämnet ger dig en procedur från början till slut som beskriver hur du implementerar webbnivån i exemplet på AWS-referensarkitektur med hjälp av en oberoende gateway för Tableau Server. Ett exempel på konfiguration med Apache som omvänd proxy finns i Bilaga – Exempel på driftsättning på webbnivå med Apache.

En exempelkonfiguration består av följande komponenter:

- Lastbalanserare för AWS-programmet
- Oberoende gateway för Tableau Server
- Mellon-autentiseringsmodul
- IdP för Okta
- SAML-autentisering

Obs! Det exempel på webbnivåkonfiguration som visas i det här avsnittet innehåller detaljerade procedurer för att distribuera programvara och tjänster från tredje part. Vi har gjort vårt bästa för att verifiera och dokumentera procedurerna för att möjliggöra webbnivåscenariot. Programvara från tredje part kan dock ändras eller så kan ditt scenario skilja sig från den referensarkitektur som beskrivs här. Se dokumentationen från tredje part för konfigurationsinformation och support som har företräde.

Linux-exemplen i detta avsnitt visar kommandon för RHEL-liknande distributioner. Mer specifikt har kommandona här utvecklats med Amazon Linux 2-distributionen. Om du kör Ubuntu-driftsättningen redigerar du kommandona på lämpligt sätt.

Driftsättning av webbnivån i det här exemplet följer en stegvis konfigurations- och verifieringsprocedur. Konfigurationen av kärnwebbnivån består av följande steg för att aktivera HTTP mellan Tableau och internet. Den oberoende gatewayen körs och konfigureras för omvänd proxy/lastbalansering bakom lastbalanseraren för AWS-programmet:

1. Förbereda miljön
2. Installera oberoende gateway
3. Konfigurera oberoende gatewayserver
4. Konfigurera lastbalanserare för AWS-program

Efter att webbnivån konfigurerats och anslutning med Tableau verifierats så konfigurerar du autentisering med den externa leverantören.

Förbereda miljön

Slutför följande uppgifter innan du driftsätter en oberoende gateway.

1. Förändringar av AWS-säkerhetsgruppen. Konfigurera den offentliga säkerhetsgruppen för att tillåta den oberoende gatewayens inkommande housekeeping-trafik (TCP 21319) från den privata säkerhetsgruppen.

2. Installera version 22.1.1 (eller senare) på Tableau Server-kluster med fyra noder enligt anvisningarna i Del 4 – Installera och konfigurera Tableau Server.

3. Konfigurera de två proxy-EC2-instanserna i den offentliga säkerhetsgruppen enligt anvisningarna i Konfigurera värddatorer.

Installera oberoende gateway

Den oberoende gatewayen för Tableau Server kräver en licens för Advanced Management.

Driftsättningen av oberoende gateway för Tableau Server består av att installera och köra .rpm-paketet och sedan konfigurera initialtillståndet. Proceduren i denna guide ger föreskrivande vägledning för driftsättning i referensarkitekturen.

Om din driftsättning skiljer sig från referensarkitekturen hänvisar vi till den grundläggande Tableau Server-dokumentationen: *Installera Tableau Server med oberoende gateway* (Linux).

> **Viktigt**: Konfigureringen av en oberoende gateway kan ge upphov till många fel. Det är mycket svårt att felsöka konfigurationsproblem i två instanser av oberoende gateway-servrar. Av denna anledning rekommenderar vi att du konfigurerar en enstaka oberoende gateway-server åt gången. När du har konfigurerat den första servern och verifierat att den fungerar konfigurerar du den andra oberoende gateway-servern.

Även om du kommer att konfigurera varje oberoende gateway-server separat, ska du köra följande installationsprocedur på båda EC2-instanserna som du installerade i den offentliga säkerhetsgruppen:

1. Kör en uppdatering för att installera de senaste Linux OS-korrigeringarna:

   ```
   sudo yum update
   ```

2. Om Apache är installerat ska du ta bort det:

   ```
   sudo yum remove httpd
   ```

3. Kopiera installationspaketet för oberoende gateway version 2022.1.1 (eller senare) från sidan Tableau-nedladdningar till den värddator som kommer att köra Tableau Server.

   På en dator som kör ett Linux RHEL-liknande operativsystem kör du till exempel:

   ```
   wget
   https://downloads.tableau.com/esdalt/2022<version>/tableau-
   server-tsig-<version>.x86_64.rpm
   ```

4. Kör installationsprogrammet. På ett Linux RHEL-liknande operativsystem kör du till exempel:

   ```
   sudo yum install <tableau-tsig-version>.x86_64.rpm
   ```

5. Ändra till katalogen `/opt/tableau/tableau_tsig/packages/scripts.<version_code>/` och kör skriptet `initialize-tsig` som finns där. Utöver flaggan `--accepteula` måste du inkludera IP-intervallet för undernäten där Tableau Server-driftsättningen körs. Använd alternativet `-c` för att ange IP-intervallet. Exemplet nedan visar kommandot med exemplet på AWS-undernät:

   ```
   sudo ./initialize-tsig --accepteula -c "ip 10.0.30.0/24
   10.0.31.0/24"
   ```

6. När initieringen är klar öppnar du filen `tsighk-auth.conf` och kopiera autentiseringshemligheten i filen. Du måste skriva in den här koden för varje oberoende gateway-instans som en del av backend Tableau Server-konfigurationen:

   ```
   sudo less /var/opt/tableau/tableau_tsig/config/tsighk-auth.conf
   ```

7. När du har kört de föregående stegen på båda oberoende gateway-instanserna förbereder du konfigurationsfilen `tsig.json`. Konfigurationsfilen består av en ”independentGateways”-matris. Matrisen innehåller konfigurationsobjekt som var och en definierar anslutningsdetaljer för en oberoende gateway-instans.

Kopiera följande JSON och anpassa den efter din distributionsmiljö. Exemplet här visar en fil för ett exempel på AWS-referensarkitektur.

JSON-exemplet nedan innehåller endast anslutningsinformation för en enstaka oberoende gateway. Senare i processen kommer du att inkludera anslutningsinformationen för den andra oberoende gateway-servern.

Spara filen som `tsig.json` för de efterföljande procedurerna.

```
{
"independentGateways": [
 {
       "id": "ip-10-0-1-169.ec2.internal",
       "host": "ip-10-0-1-169.ec2.internal",
       "port": "21319",
       "protocol" : "http",
       "authsecret": "13660-27118-29070-25482-9518-22453"
       }]
 }
```

- `"id"` – det privata DNS-namnet för AWS EC2-instansen som kör den oberoende gatewayen.
- `"host"` – samma som `"id"`.
- `"port"` – housekeeping-porten, som standard: `"21319"`.
- `"protocol"` – protokollet för klienttrafik. Lämna detta som `http` för den initiala konfigurationen.
- `"authsecret"` – hemligheten som du kopierade i föregående steg.

Oberoende gateway: direkt kontra omdirigerad anslutning

Innan du fortsätter måste du bestämma vilket anslutningsschema du ska konfigurera i din driftsättning: direkt eller omdirigerad anslutning. Varje alternativ beskrivs kortfattat här nedan, tillsammans med relevanta beslutsdatapunkter.

Omdirigerad anslutning: Du kan konfigurera den oberoende gatewayen att skicka vidare klientkommunikation över en enda port till gateway-processen på Tableau Server. Vi kallar detta för en *omdirigerad anslutning*:

- Omdirigeringsprocessen resulterar i ett extra hopp från den oberoende gatewayen till gateway-backendprocessen på Tableau Server. Det extra hoppet försämrar prestandan jämfört med en direkt anslutning.
- TLS stöds för omdirigeringsläge. All kommunikation i omdirigeringsläge är begränsad till ett enda protokoll (HTTP eller HTTPS) och kan därför krypteras och autentiseras med TLS.

Direkt anslutning: Den oberoende gatewayen kan kommunicera direkt med serverprocesserna i Tableau Server över flera portar. Vi kallar den här typen av kommunikation för *direkt anslutning*:

- Eftersom anslutningen är direkt till backend Tableau Server förbättras klientprestandan markant jämfört med alternativet med omdirigeringsanslutning.
- Kräver att över 16 portar öppnas från offentliga till privata undernät för direkt processkommunikation från den oberoende gatewayen till Tableau Server-datorer.
- TLS stöds ännu inte på processerna från oberoende gatewayer till Tableau Server.

Konfigurera omdirigeringsanslutning

För att köra TLS mellan Tableau Server och en oberoende gateway måste du utföra konfigurationen med en omdirigeringsanslutning. Exempelscenarierna i EDG är konfigurerade med omdirigeringsanslutning.

1. Kopiera `tsig.json` till nod 1 i din Tableau Server-driftsättning.
2. Kör följande kommandon på nod 1 för att aktivera den oberoende gatewayen.

```
tsm stop
tsm configuration set -k gateway.tsig.proxy_tls_optional -v
none
tsm pending-changes apply
tsm topology external-services gateway enable -c tsig.json
tsm start
```

Konfigurera direkt anslutning

Eftersom direkt anslutning inte stöder TLS rekommenderar vi att du endast konfigurerar denna typ av anslutning om du kan säkra all nätverkstrafik på annat sätt. För att köra TLS mellan Tableau Server och en oberoende gateway måste du utföra konfigurationen med en omdirigeringsanslutning. Exempelscenarierna i EDG är konfigurerade med omdirigeringsanslutning.

Om du konfigurerar en oberoende gateway för direkt anslutning till Tableau Server, måste du aktivera konfigurationen för att utlösa kommunikationen. Efter att Tableau Server kommunicerat med den oberoende gatewayen kommer protokollmålen att fastställas. Du måste sedan hämta `proxy_targets.csv` från datorn med den oberoende gatewayen och öppna motsvarande portar från den offentliga till den privata säkerhetsgruppen i AWS.

1. Kopiera `tsig.json` till nod 1 i din Tableau Server-driftsättning.
2. Kör följande kommandon på nod 1 för att aktivera den oberoende gatewayen.

   ```
   tsm stop
   tsm topology external-services gateway enable -c tsig.json
   tsm start
   ```

3. På datorn med den oberoende gatewayen kör du följande kommando för att se de portar som Tableau Server-klustret använder:

   ```
   less /var/opt/tableau/tableau_tsig/config/httpd/proxy_
   targets.csv
   ```

4. Konfigurera AWS-säkerhetsgrupper. Lägg till TCP-portarna som anges i `proxy_targets.csv` för att tillåta kommunikation från den offentliga säkerhetsgruppen till den privata säkerhetsgruppen.

   Vi rekommenderar att konfigurationen av ingångsportarna automatiseras då portarna kan ändras om driftsättningen av Tableau Server ändras. Att lägga till noder eller

konfigurera processer på nytt i Tableau Server-driftsättningen utlöser ändringar av portåtkomsten som krävs av oberoende gateway.

Verifiering: bastopologikonfiguration

Du borde kunna komma åt adminsidan för Tableau Server genom att gå till `http://<gateway-public-IP-address>`.

Om inloggningssidan för Tableau Server inte läses in, eller om Tableau Server inte startar, kan du följa dessa felsökningssteg:

Nätverk:

- Verifiera anslutningen mellan Tableau-driftsättningen och den oberoende gateway-instansen genom att köra kommandot `wget` från Tableau Server, nod 1: wget http://<internal IP address of Independent Gateway> :21319, till exempel:

  ```
  wget http://ip-10-0-1-38:21319
  ```

  Om anslutningen nekas eller misslyckas, ska du verifiera att den offentliga säkerhetsgruppen är konfigurerad för att tillåta den oberoende gatewayens housekeeping-trafik (TCP 21319) från den privata säkerhetsgruppen.

  Om säkerhetsgruppen är korrekt konfigurerad, ska du kontrollera att du angav rätt IP-adresser eller IP-intervall under initieringen av den oberoende gatewayen. Du kan visa och ändra denna konfiguration i filen `environment.bash` som finns på`/etc/opt/tableau/tableau_tsig/environment.bash`. Om du gör en ändring i den här filen behöver du starta om tsig-http-tjänsten enligt beskrivningen nedan.

På proxy 1-värden:

1. Skriv över httpd.conf-filen med oberoende gateway-stubfilen:

   ```
   cp /var/opt/tableau/tableau_tsig/config/httpd.conf.stub
   /var/opt/tableau/tableau_tsig/config/httpd.conf
   ```

2. Starta om tsig-httpd som ett första felsökningssteg:

```
sudo su - tableau-tsig
systemctl --user restart tsig-httpd
exit
```

Nod 1 i Tableau

- Dubbelkolla filen `tsig.json`. Om du hittar fel ska du åtgärda dem och sedan köra `tsm topology external-services gateway update -c tsig.json`.
- Om du kör direkt anslutning, ska du verifiera att TCP-portarna som anges i `proxy_targets.csv` är konfigurerade som ingångsportar från offentliga till privata säkerhetsgrupper.

Konfigurera lastbalanserare för AWS-program

Konfigurera belastningsutjämnaren som en HTTP-lyssnare. Här beskrivs hur du lägger till en belastningsutjämnare i AWS:

Steg 1: Skapa målgrupp

Målgruppen är en AWS-konfiguration där de EC2-instanser som körs för dina proxyservrar definieras. Dessa är målen för LBS-trafik.

1. EC2 >**Målgrupper** > **Skapa målgrupp**

2. På sidan Skapa:

    - Ange ett namn på målgruppen, till exempel `TG-internal-HTTP`
    - Måltyp: instanser
    - Protokoll: HTTP
    - Port: 80
    - VPC: Välj VPC
    - Under **Hälsokontroller** > **Avancerade inställningar för hälsokontroller** > **Framgångskoder** lägger du till kodlistan enligt följande: `200,303`.
    - Klicka på **Skapa**

3. Välj den målgrupp som du just skapade och klicka sedan på fliken **Mål**:

- Klicka på **Redigera**.
- Välj de EC2-instanser (eller en instans om du konfigurerar en i taget) som kör proxyprogram och klicka sedan på **Lägg till bland registrerade**.
- Klicka på **Spara**.

Steg 2: Starta guiden för belastningsutjämnaren

1. EC2 > **Belastningsutjämnare** > **Create Load Balancer** (Skapa belastningsutjämnare)
2. Skapa en belastningsutjämnare för program på sidan ”Välj typ av belastningsutjämnare”.

Obs! Gränssnittet som visas för konfigurering av belastningsutjämnaren är inte samma på alla AWS-datacenter. Via stegen i Guidekonfiguration nedan får du åtkomst till AWS-konfigurationsguiden som börjar med **steg 1 Konfigurera belastningsutjämnare**.

Om alla konfigurationer visas på samma sida i datacentret där knappen **Skapa belastningsutjämnare** visas längst ned följer du proceduren ”Konfiguration för enskild sida” nedan.

Guidekonfiguration

1. Sidan **Konfigurera belastningsutjämnare**:
    - Ange namn
    - Schema: internetanpassat (standard)
    - IP-adresstyp: ipv4 (standard)
    - Lyssnare (lyssnare och dirigering):
        a. Lämna HTTP-standardlyssnaren
        b. Klicka på **Lägg till lyssnare** och lägg till `HTTPS:443`

- VPC: välj den VPC där du har installerat allt
- Tillgänglighetszoner:
  - Välj **a** och **b** för dina datacenterregioner
  - I varje motsvarande listruta väljer du det offentliga undernätet (där dina proxyservrar finns).
- Klicka på **Configure Security Settings** (Konfigurera säkerhetsinställningar)

2. Sidan **Konfigurera säkerhetsinställningar**
   - Ladda upp ditt offentliga SSL-certifikat.
   - Klicka på **Next: Configure Security Groups** (Nästa: Konfigurera säkerhetsgrupper).
3. Sidan **Konfigurera säkerhetsgrupper**:
   - Välj den offentliga säkerhetsgruppen. Om standardvalet för säkerhetsgrupp väljs, ska valet rensas.
   - Klicka på **Next: Configure Routing** (Nästa: Konfigurera dirigering).
4. Sidan **Configure Routing** (Konfigurera dirigering).
   - Målgrupp: Befintlig målgrupp.
   - Namn: Välj den målgrupp som du skapade tidigare.
   - Klicka på **Next: Register Targets** (Nästa: Registrera mål).
5. Sidan **Register Targets** (Registrera mål)
   - Du borde se de två proxyserverinstanserna som du konfigurerade tidigare.
   - Klicka på **Next: Review** (Nästa: Granska).
6. Sidan **Review** (Granska)

   Klicka på **Skapa**.

Konfiguration för enskild sida

Grundläggande konfiguration

- Ange namn
- Schema: internetanpassat (standard)
- IP-adresstyp: ipv4 (standard)

Nätverkskartläggning

- VPC: välj den VPC där du har installerat allt
- Kartläggningar:
  - Välj tillgänglighetszoner **a** och **b** (eller liknande) för dina datacenterregioner
  - I varje motsvarande listruta väljer du det offentliga undernätet (där dina proxyservrar finns).

Säkerhetsgrupper

Välj den offentliga säkerhetsgruppen. Om standardvalet för säkerhetsgrupp väljs, ska valet rensas.

Lyssnare och dirigering

- Lämna HTTP-standardlyssnaren. För **Standardåtgärd** anger du Målgruppen som du ställde in tidigare.
- Klicka på **Lägg till lyssnare** och lägg till `HTTPS:443`. För **Standardåtgärd** anger du Målgruppen som du ställde in tidigare.

Skydda lyssnarinställningarna

- Ladda upp ditt offentliga SSL-certifikat.

Klicka på **Create load balancer** (Skapa belastningsutjämnare).

Steg 3: Aktivera varaktighet

1. När belastningsutjämnaren har skapats ska varaktigheten aktiveras i målgruppen.
   - Öppna sidan AWS-målgrupp (**EC2** > **Belastningsutjämning** > **Målgrupper**) och välj den målgruppsinstans som du just konfigurerade. I **Åtgärdsmenyn** väljer du **Redigera attribut**.
   - På sidan **Redigera attribut** väljer du **Varaktighet**, anger en varaktigheten `1 day` och trycker sedan på **Spara ändringar**.

2. Aktivera varaktighet för belastningsutjämnaren för HTTP-lyssnaren. Välj den belastningsutjämnare som du just konfigurerade och klicka sedan på fliken **Lyssnare**:

    - För **HTTP:80** klickar du på **Visa/redigera regler**. Öppna sidan **Regler** och klicka därefter på redigeringsikonen (en gång högst upp på sidan och sedan igen bredvid regeln) för att redigera regeln. Ta bort befintlig THEN-regel och ersätt genom att klicka på **Lägg till åtgärd** > **Vidarebefordra till ...**. Ange den målgrupp som du har skapat i återstående DÅ-konfiguration. Aktivera varaktighet under Varaktighet på gruppnivå och ange varaktigheten till 1 dag. Spara inställningen och klicka sedan på **Uppdatera**.

Steg 4: Ställ in tidsgräns för inaktivitet för belastningsutjämnaren

Uppdatera tidsgränsen för inaktivitet till 400 sekunder för belastningsutjämnaren.

Välj den belastningsutjämnare du har konfigurerat för den här driftsättningen och klicka sedan på **Åtgärder** > **Redigera attribut**. Ställ in **tidsgränsen för inaktivitet** på `400` sekunder och klicka sedan på **Spara**.

Steg 5: Verifiera LBS-anslutning

Öppna sidan AWS-belastningsutjämnare (**EC2**> **Belastningsutjämnare**) och välj den belastningsutjämnare som du just konfigurerade.

Kopiera DNS-namnet som visas i **Beskrivning** och klistra in i webbläsaren för att komma åt inloggningssidan för Tableau Server.

Om ett 500-nivåfel uppstår måste du starta om proxyservrarna.

Uppdatera DNS med den offentliga Tableau-URL:en

Använd domänens DNS-zonnamn från beskrivningen för AWS-belastningsutjämnaren för att skapa ett CNAME-värde i din DNS. Trafik till din URL (tableau.example.com) borde skickas till ditt offentliga AWS DNS-namn.

Verifiera anslutning

Efter att dina DNS-uppdateringar slutförts borde du kunna gå till inloggningssidan för Tableau Server genom att ange din offentliga URL, till exempel `https://tableau.example.com`.

Exempel på autentiseringskonfiguration: SAML med extern IdP

Följande exempel beskriver hur du installerar och konfigurerar SAML med IdP för Okta och Mellon-autentiseringsmodulen för en Tableau-driftsättning som kör AWS-referensarkitekturen.

Det här exemplet fortsätter från föregående avsnitt och förutsätter att du konfigurerar en enda oberoende gateway åt gången.

Det här exemplet beskriver hur du konfigurerar Tableau Server och Apache-proxyservrar över HTTP. Okta skickar begäranden till AWS-belastningsutjämnaren över HTTPS men all intern trafik går över HTTP. När du konfigurerar för det här scenariot bör du vara medveten om HTTP- kontra HTTPS-protokollen när du anger URL-strängar.

Det här exemplet använder sig av Mellon som tjänsteleverantörsmodul för förautentisering på de oberoende gateway-servrarna. Med den här konfigurationen ser du till att endast autentiserad trafik ansluter till Tableau Server, som även agerar som tjänstleverantör med IdP för Okta. Därmed behöver du konfigurera två IdP-applikationer: en för Mellon-tjänstleverantören och en för Tableau-tjänstleverantören.

Skapa ett Tableau-administratörskonto

Ett vanligt misstag när man konfigurerar SAML är att inte skapa ett administratörskonto på Tableau Server innan SSO aktiveras.

Första steget är att skapa ett konto på Tableau Server med rollen som serveradministratör. För exemplet med Okta-scenariot måste användarnamnet vara i ett giltigt e-

postadressformat, såsom användare@exempel.com. Du måste ange ett lösenord för den här användaren. Lösenordet kommer dock inte att användas efter att SAML har konfigurerats.

Konfigurera Okta-program med förautentisering

Scenariot från slutpunkt till slutpunkt som beskrivs i det här avsnittet kräver två Okta-program:

- Okta-program med förautentisering
- Okta Tableau Server-program

Var och en av dessa program är associerade med olika metadata som behöver konfigureras på den omvända proxyn och Tableau-servern, respektive.

Den här proceduren beskriver hur man skapar och konfigurerar Okta-program med förautentisering. Längre fram i detta ämne kommer du att skapa Okta Tableau Server-programmet. Se Oktas webbplats för utvecklare för ett kostnadsfritt Okta-testkonto med begränsade användare.

Skapa en SAML-appintegrering för Mellon-tjänsteleverantör med förautentisering.

1. Öppna Oktas administrationsöversikt > **Program** > **Skapa appintegrering**.
2. På sidan **Skapa en ny appintegrering** kan du välja **SAML 2.0** och sedan klicka på **Nästa**.
3. På fliken **Allmänna inställningar** ska du ange ett appnamn, såsom `Tableau Pre-Auth`, och sedan klicka på **Nästa**.
4. På fliken **Konfigurera SAML**:
    - URL för enkel inloggning. Det sista elementet av sökvägen i URL:en för enkel inloggning kallas för `MellonEndpointPath` i konfigurationsfilen `mellon.conf` som följer senare i den här proceduren. Du kan ange valfri slutpunkt. I det här exemplet är slutpunkten `sso`. Det sista elementet `postResponse` krävs : `https://tableau.example.com/sso/postResponse`.

- Klicka ur kryssrutan: **Använd detta för URL för mottagaren och destinationen**.
- URL för mottagaren: Samma som URL för SSO, men med HTTP. Till exempel `http://tableau.example.com/sso/postResponse`.
- Mål-URL: samma som URL för enkel inloggning, men med HTTP. Till exempel `http://tableau.example.com/sso/postResponse`.
- Audience URI (SP Entity ID). Till exempel `https://tableau.example.com`.
- Format på namn-ID: `EmailAddress`
- Användarnamn i programmet: `Email`
- Attribututlåtanden: namn = `mail`; namnformat = `Unspecified`; värde = `user.email`.

Klicka på **Nästa**.

5. Välj följande på fliken **Feedback**:

   - **Jag är Okta-kund och lägger till ett internt program**
   - **Det här är ett internt program som vi har skapat**
   - Klicka på **Slutför**.

6. Skapa metadatafilen för tjänsteleverantörens förautentisering:

   - I Okta: **Program** > **Program** > Ditt nya program (t.ex. `Tableau Pre-Auth`) > **Logga in**
   - Bredvid **SAML-signeringscertifikat** klickar du på **Visa anvisningar om SAML-konfiguration**.
   - På sidan **Så här konfigurerar du SAML 2.0 för <pre-auth>-program** rullar du nedåt till delen **Valfritt** , **Uppge följande IdP-metadata för serviceleverantören**.
   - Kopiera innehållet i XML-fältet och spara det i en fil med namnet `pre-auth_idp_metadata.xml`.

7. (Valfritt) Konfigurera multifaktorautentisering:

   - I Okta: **Program** > **Program** > Ditt nya program (t.ex. `Tableau Pre-Auth`) > **Logga in**
   - Klicka på **Lägg till regel** under **Inloggningspolicy**.

- Ange ett namn och de olika MFA-alternativen i **Programmets inloggningsregel**. Du kan lämna alla alternativ som standard för att testa funktionaliteten. Under **Åtgärder** måste du dock välja **Fråga efter faktor** och sedan ange hur ofta användare måste logga in. Klicka på **Spara**.

Skapa och tilldela Okta-användare

1. Skapa en användare, i Okta, med samma användarnamn som skapades i Tableau (användare@example.com): **Katalog** > **Personer** > **Lägg till person**.
2. Tilldela dett nya Okta-programmet till den personen när användaren har skapats: Klicka på användarnamnet och tilldela sedan programmet i **Tilldela program**.

Installera Mellon för förautentisering

Det här exemplet använder mod_auth_mellon, en populär öppen källkodsmodul. Vissa driftsättningar på Linux paketerar föråldrade mod_auth_mellon-versioner från en äldre lagringsplats. Dessa föråldrade versioner kan innehålla okända säkerhetsbrister eller funktionsproblem. Om du väljer att använda mod_auth_mellon ska du kontrollera att du använder en aktuell version.

Mod_auth_mellon-modulen är programvara från tredje part. Vi har gjort vårt bästa för att verifiera och dokumentera procedurerna för att möjliggöra det här scenariot. Programvara från tredje part kan dock ändras eller så kan ditt scenario skilja sig från den referensarkitektur som beskrivs här. Se dokumentationen från tredje part för konfigurationsinformation och support som har företräde.

1. Installera en aktuell version av Mellon-autentiseringsmodulen på den aktiva EC2-instansen som kör den oberoende gatewayen.

2. Skapa katalogen `/etc/mellon`:

   ```
   sudo mkdir /etc/mellon
   ```

Konfigurerar Mellon som förautentiseringsmodul

Kör den här proceduren på den första instansen av den oberoende gatewayen.

Du måste ha en kopia av `pre-auth_idp_metadata.xml`-filen som du skapade från Okta-konfigurationen.

1. Ändra katalog:

   ```
   cd /etc/mellon
   ```

2. Skapa metadata för tjänstleverantören. Kör skriptet `mellon_create_metadata.sh`. Du måste inkludera entitets-ID och retur-URL för din organisation i kommandot.

   Retur-URL kallas för *URL för enkel inloggning* i Okta. Det slutliga elementet för sökvägen i retur-URL:en kallas för `MellonEndpointPath` i konfigurationsfilen `mellon.conf` som kommer senare i den här proceduren. I det här exemplet anger vi `sso` som slutpunktssökväg.

   Exempel:

   ```
   sudo /usr/libexec/mod_auth_mellon/mellon_create_metadata.sh
   https://tableau.example.com "https://tableau.example.com/sso"
   ```

   Skriptet returnerar tjänstleverantörens certifikat, nyckel och metadatafiler.

3. Byt namn på tjänstleverantörens filer i katalogen `mellon` för ökad läsbarhet. Vi kallar dessa filer följande namn i dokumentationen:

   ```
   sudo mv *.key mellon.key
   sudo mv *.cert mellon.cert
   sudo mv *.xml sp_metadata.xml
   ```

4. Kopiera `pre-auth_idp_metadata.xml`-filen till samma katalog.

5. Ändra äganderätt och behörigheter för alla filer i `/etc/mellon`-katalogen:

   ```
   sudo chown tableau-tsig mellon.key
   sudo chown tableau-tsig mellon.cert
   sudo chown tableau-tsig sp_metadata.xml
   ```

```
sudo chown tableau-tsig pre-auth_idp_metadata.xml
sudo chmod +r * mellon.key
sudo chmod +r * mellon.cert
sudo chmod +r * sp_metadata.xml
sudo chmod +r * pre-auth_idp_metadata.xml
```

6. Skapa katalogen `/etc/mellon/conf.d`:

   ```
   sudo mkdir /etc/mellon/conf.d
   ```

7. Skapa filen `global.conf` i katalogen `/etc/mellon/conf.d`.

   Kopiera filinnehållet som visas nedan, men uppdatera `MellonCookieDomain` med ditt rotdomännamn. Om domännamnet för Tableau till exempel är `tableau.example.com` ska du ha `example.com` som rotdomän.

   ```
   <Location "/">
   AuthType Mellon
   MellonEnable auth
   Require valid-user
   MellonCookieDomain <root domain>
   MellonSPPrivateKeyFile /etc/mellon/mellon.key
   MellonSPCertFile /etc/mellon/mellon.cert
   MellonSPMetadataFile /etc/mellon/sp_metadata.xml
   MellonIdPMetadataFile /etc/mellon/pre-auth_idp_metadata.xml
   MellonEndpointPath /sso
   </Location>

   <Location "/tsighk">
   MellonEnable Off
   </Location>
   ```

8. Skapa filen `mellonmod.conf` i katalogen `/etc/mellon/conf.d`.

   Den här filen innehåller ett enda direktiv som anger platsen för filen `mod_auth_mellon.so`. Platsen i exemplet här är standardplatsen för filen. Verifiera att filen finns

på den här platsen, eller ändra sökvägen i detta direktiv till att matcha den faktiska platsen för `mod_auth_mellon.so`:

```
LoadModule auth_mellon_module /usr/lib64/httpd/modules/mod_
auth_mellon.so
```

Skapa Tableau Server-applikation i Okta

1. På instrumentpanelen för Okta: **Program** > **Program** > **Browse App Catalog** (Bläddra i appkatalog).
2. Sök efter `Tableau` i **Browse App Integration Catalog** (Bläddra i appintegreringskatalog), välj panelen för Tableau Server och klicka på **Lägg till**.
3. På **Add Tableau Server** (Lägg Tableau Server) > **General Settings** (Allmänna inställningar) anger du en etikett och klickar sedan på **Nästa**.
4. Välj **SAML 2.0** i Sign-On Options (Inloggningsalternativ) och rulla ned till Advanced Sign-on Settings (Avancerade inloggningsinställningar):
   - **SAML Entity ID** (Entitets-ID för SAML): Ange den offentliga URL:en, t.ex. https://tableau.example.com.
   - **Application user name format** (Format för användarnamn för programmet): E-postadress
5. Starta en webbläsare genom att klicka på länken **Identity Provider metadata** (Metadata för identitetsprovider). Kopiera webbläsarlänken. Det här är länken du använder när du konfigurerar Tableau i stegen som följer.
6. Klicka på **Klart**.
7. Tilldela användaren (användare@företag.com) den nya Tableau Server Okta-appen: Klicka på användarnamnet och tilldela sedan programmet i **Assign Application** (Tilldela program).

Ställa in konfiguration av autentiseringsmodul på Tableau Server

Kör följande kommandon på Tableau Server, nod 1. Dessa kommandon anger filplatserna för Mellon-konfigurationsfilerna på den fjärranslutna datorn med den oberoende gatewayen. Dubbelkolla att de filsökvägar som anges i dessa kommandon verkligen går till sökvägarna och filplatsen på den fjärranslutna datorn.

```
tsm configuration set -k gateway.tsig.authn_module_block -v
"/etc/mellon/conf.d/mellonmod.conf" --force-keys
tsm configuration set -k gateway.tsig.authn_global_block -v
"/etc/mellon/conf.d/global.conf" --force-keys
```

För att minska stilleståndstiden ska du inte tillämpa ändringar förrän du har aktiverat SAML enligt beskrivningen i nästa avsnitt.

Aktivera SAML på Tableau Server för IdP

Kör den här proceduren på Tableau Server-nod 1.

1. Hämta metadata för Tableau Server-programvaran från Okta. Använd länken som du sparade från föregående procedur.

   ```
   wget https://dev-
   66144217.okta.com/app/exk1egxgt1fhjkSeS5d7/sso/saml/metadata -O
   idp_metadata.xml
   ```

2. Kopiera ett TLS-certifikat och den relaterade nyckelfilen till Tableau Server. Nyckelfilen måste vara en RSA-nyckel. Mer information om SAML-certifikat och IdP-krav finns i *SAML-krav* (Linux).

   För att underlätta certifikathanteringen och driftsättningen, och som en rekommenderad säkerhetsåtgärd, bör du använda certifikat genererade av någon av de stora betrodda certifikatutfärdarna (CA). Du kan också generera självsignerade certifikat eller använda certifikat från en PKI för TLS.

   Om du inte har något TLS-certifikat kan du skapa ett självsignerat certifikat med hjälp av den inbäddade proceduren nedan.

Skapa ett självsignerat certifikat

Kör den här proceduren på Tableau Server-nod 1.

a. Generera signeringsnyckel för rotcertifikatutfärdare (CA, Certificate Authority):

```
openssl genrsa -out rootCAKey-saml.pem 2048
```

b. Skapa certifikat för rot-CA:

```
openssl req -x509 -sha256 -new -nodes -key rootCAKey-
saml.pem -days 3650 -out rootCACert-saml.pem
```

Du uppmanas att ange värden i certifikatfälten. Exempel:

```
Country Name (2 letter code) [XX]:US
State or Province Name (full name) []:Washington
Locality Name (eg, city) [Default City]:Seattle
Organization Name (eg, company) [Default Company
Ltd]:Tableau
Organizational Unit Name (eg, section) []:Operations
Common Name (eg, your name or your server's hostname)
[]:tableau.example.com
Email Address []:example@tableau.com
```

c. Skapa certifikatet och relaterad nyckel (`server-saml.csr` och `server-saml.key` i nedanstående exempel). Certifikatmottagarens namn måste stämma överens med det offentliga värdnamnet för Tableau-värden. Certifikatmottagarens namn anges med alternativet `-subj` i formatet `"/CN=<host-name>"`, till exempel:

```
openssl req -new -nodes -text -out server-saml.csr -keyout
server-saml.key -subj "/CN=tableau.example.com"
```

d. Signera det nya certifikatet med CA-certifikatet som du skapade ovan. Följande kommando skapar även certifikatet i `crt`-format:

```
openssl x509 -req -in server-saml.csr -days 3650 -CA
rootCACert-saml.pem -CAkey rootCAKey-saml.pem -
```

```
CAcreateserial -out server-saml.crt
```

e. Konvertera nyckelfilen till RSA. Tableau behöver en RSA-nyckelfil för SAML. Kör följande kommando för att konvertera nyckeln:

```
openssl rsa -in server-saml.key -out server-saml-rsa.key
```

3. Konfigurera SAML. Kör följande kommando och ange ditt enhets-ID och din retur-URL, samt sökvägarna till metadatafilen, certifikatfilen och nyckelfilen:

```
tsm authentication saml configure --idp-entity-id
"https://tableau.example.com" --idp-return-url
"https://tableau.example.com" --idp-metadata idp_metadata.xml -
-cert-file "server-saml.crt" --key-file "server-saml-rsa.key"

tsm authentication saml enable
```

4. Om din organisation använder Tableau Desktop 2021.4 eller senare måste du köra följande kommando för att aktivera autentisering via omvända proxyservrar.

   Tableau Desktop-versioner 2021.2.1 – 2021.3 kommer att fungera utan att köra detta kommando, förutsatt att din förautentiseringsmodul (t.ex. Mellon) har konfigurerats för att tillåta att domäncookies bevaras på toppnivå.

```
tsm configuration set -k features.ExternalBrowserOAuth -v false
```

5. Tillämpa konfigurationsändringar:

```
tsm pending-changes apply
```

Starta om tsig-httpd-tjänsten

Om Tableau Server-driftsättningen tillämpar ändringar loggar du tillbaka in på Tableau Server-datorn med den oberoende gatewayen och kör följande kommandon för att starta om tsig-httpd-tjänsten:

```
sudo su - tableau-tsig
systemctl --user restart tsig-httpd
exit
```

Validera SAML-funktion

Validera SAML-funktion från slutpunkt till slutpunkt genom att logga in på Tableau Server med den offentliga URL:en (t.ex., https://tableau.example.com) med det Tableau-administratörskonto som du skapade i början av den här proceduren.

Om TSM inte startar (d.v.s. ”gateway-fel”) eller om du får webbläsarfel när du försöker ansluta, kan du ta en titt i Felsöka oberoende gateway för Tableau Server.

Konfigurera autentiseringsmodulen på den andra oberoende gateway-instansen

När du har konfigurerat den första instansen av oberoende gateway driftsätter du den andra instansen. Exemplet här är den sista processen för att installera AWS-/Mellon-/Okta-scenariot som beskrivs i detta ämne. Proceduren förutsätter att du redan har installerat den andra oberoende gateway-instansen enligt beskrivningen i avsnittet tidigare (Installera oberoende gateway).

För att driftsätta den andra oberoende gatewayen behöver du följa stegen nedan:

1. På den andra oberoende gateway-instansen ska du installera Mellon auth-modulen.

   Konfigurera inte Mellon auth-modulen som beskrevs längre fram i det här avsnittet. Istället måste du klona konfigurationen enligt beskrivningen i de efterföljande stegen.

2. På den konfigurerade (första) oberoende gateway-instansen:

   Ta en tar-kopia av den befintliga Mellon-konfigurationen. Tar-säkerhetskopieringen kommer att bevara kataloghierarkin och alla behörigheter. Kör följande kommandon:

   ```
   cd /etc
   ```

```
sudo tar -cvf mellon.tar mellon
```

Kopiera `mellon.tar` till den andra oberoende gateway-instansen.

3. På den andra oberoende gateway-instansen:

   Extrahera (packa upp) tar-filen till den andra instansen i `/etc`-katalogen. Kör följande kommandon:

   ```
   cd /etc
   ```

   ```
   sudo tar -xvf mellon.tar
   ```

4. På nod 1 i Tableau Server-driftsättningen uppdaterar du anslutningsfilen (`tsig.json`) med anslutningsinformationen från den andra oberoende gatewayen. Du måste hämta autentiseringsnyckeln enligt beskrivningen i följande föregående avsnitt (Installera oberoende gateway).

   Ett exempel på anslutningsfil (`tsig.json`) visas här:

   ```
   {
   "independentGateways": [
    {
      "id": "ip-10-0-1-169.ec2.internal",
      "host": "ip-10-0-1-169.ec2.internal",
      "port": "21319",
      "protocol" : "http",
      "authsecret": "13660-27118-29070-25482-9518-22453"
    },
    {
      "id": "ip-10-0-2-230.ec2.internal",
      "host": "ip-10-0-2-230.ec2.internal",
      "port": "21319",
      "protocol" : "http",
      "authsecret": "9055-27834-16487-27455-30409-7292"
    }]
    }
   ```

5. På nod 1 i Tableau Server-driftsättningen kör du följande kommandon för att uppdatera konfigurationen:

   ```
   tsm stop

   tsm topology external-services gateway update -c tsig.json

   tsm start
   ```

6. På båda oberoende gateway-instanserna: När Tableau Server startar, startar du om `tsig-httpd`-processen:

   ```
   sudo su - tableau-tsig

   systemctl --user restart tsig-httpd

   exit
   ```

7. I AWS **EC2 > Målgrupper**: Uppdatera målgruppen för att inkludera EC2-instansen som kör den andra oberoende gateway-instansen.

   Välj den målgrupp som du just skapade och klicka sedan på fliken Mål:

   - Klicka på **Redigera**.
   - Välj EC2-instansen för datorn med den andra oberoende gatewayen och klicka sedan på **Lägg till i registrerade**. Klicka sedan på **Spara**.

Del 6 - Konfigurera efter installation

Konfigurera SSL/TLS från belastningsutjämnare till Tableau Server

Vissa organisationer kräver en komplett (”end-to-end”) krypteringskanal från klient till backend-tjänst. Standardreferensarkitekturen som beskrivs under denna punkt specificerar SSL från klienten till den belastningsutjämnare som körs på webbnivån i din organisation.

Det här avsnittet beskriver hur du konfigurerar SSL/TLS för Tableau Server och den oberoende gatewayen i exemplet på AWS-referensarkitektur. Ett konfigurationsexempel som beskriver hur man konfigurerar SSL/TLS på Apache i AWS-referensarkitektur finns i Exempel: Konfigurera SSL/TLS i AWS-referensarkitektur.

För närvarande stöds inte TLS på Tableau Server-backendprocesserna som körs i intervallet 8000–9000. För att aktivera TLS måste du konfigurera en oberoende gateway med en reläanslutning till Tableau Server.

Den här proceduren beskriver hur du aktiverar och konfigurerar TLS på en oberoende gateway till Tableau Server och Tableau Server till den oberoende gatewayen. Proceduren krypterar relätrafiken över HTTPS/443 och rensningstrafiken över HTTPS/21319.

Linux-procedurerna i detta exempel visar kommandon för RHEL-liknande distributioner. Mer specifikt har kommandona här utvecklats med Amazon Linux 2-distributionen. Om du kör Ubuntu-driftsättningen redigerar du kommandona på lämpligt sätt.

Den här vägledningen är föreskrivande för det specifika exemplet på AWS-referensarkitektur som presenteras i den här guiden. Därför ingår inte valfria konfigurationer. Fullständig referensdokumentation finns i *Konfigurera TLS på oberoende gateway* (Linux).

Innan du konfigurerar TLS

Utför TLS-konfigurationerna utanför kontorstid. Konfigurationen kräver minst en omstart av Tableau Server. Om du kör en fullständig driftsättning av referensarkitektur med fyra noder kan omstarten ta ett tag.

- Verifiera att klienter kan ansluta till Tableau Server över HTTP. Konfigurering av TLS med oberoende gateway är en process i flera steg och kan kräva felsökning. Därför rekommenderar vi att du börjar med en fullt fungerande Tableau Server-driftsättning innan du konfigurerar TLS.

- Samla ihop TLS/SSL-certifikat, nycklar och relaterade resurser. Du behöver SSL-certifikat för oberoende gateways och för Tableau Server. För att underlätta certifikathanteringen och driftsättningen, och som en rekommenderad säkerhetsåtgärd, bör du använda certifikat genererade av någon av de stora betrodda certifikatutfärdarna (CA). Du kan också generera självsignerade certifikat eller använda certifikat från en PKI för TLS.

  Exempelkonfigurationen i det här ämnet använder följande resursnamn som illustration:
  - `tsig-ssl.crt`: TLS/SSL-certifikatet för oberoende gateway.
  - `tsig-ssl.key`: Den privata nyckeln för `tsig-ssl.crt` på oberoende gateway.
  - `ts-ssl.crt`: TLS/SSL-certifikatet för Tableau Server.
  - `ts-ssl.key`: Den privata nyckeln för `tsig-ssl.crt` på Tableau Server.
  - `tableau-server-CA.pem`: Rotcertifikatet för den CA som genererar certifikaten för Tableau Server-datorerna. Det här certifikatet krävs vanligtvis inte om du använder certifikat från större betrodda tredje parter.
  - `rootTSIG-CACert.pem`: Rotcertifikatet för den CA som genererar certifikaten för datorerna med oberoende gateway. Det här certifikatet krävs vanligtvis inte om du använder certifikat från större betrodda tredje parter.
  - Det krävs även andra certifikat och nyckelfilsresurser för SAML som beskrivs i del 5 av den här guiden.

- Om din implementering kräver att en certifikatkedjefil används, se kunskapsbasartikeln Konfigurera TLS på oberoende gateway när du använder ett certifikat som har en certifikatkedja.

- Verifiera att du har tillgång till IdP. Om du använder en IdP för autentisering behöver du sannolikt göra ändringar i mottagaren och destinationsadresserna på IdP:n efter att du har konfigurerat SSL/TLS.

Konfigurera datorer med oberoende gateway för TLS

Det kan uppstå många fel när TLS konfigureras. Eftersom felsökning över två instanser av en oberoende gateway kan vara tidskrävande rekommenderar vi att du aktiverar och konfigurerar TLS på EDG-driftsättningen med bara en oberoende gateway. När du har validerat att TLS fungerar över hela driftsättningen konfigurerar du den andra datorn med oberoende gateway.

Steg 1: Distribuera certifikat och nycklar till dator med oberoende gateway

Du kan distribuera resurserna till vilken katalog som helst så länge som tsig-httpd-användaren har läsåtkomst till filerna. Sökvägarna till de här filerna hänvisas till i andra procedurer. Vi kommer att använda exempelsökvägarna under `/etc/ssl`, som visas nedan, genom hela ämnet.

1. Skapa katalog för privat nyckel:

   ```
   sudo mkdir -p /etc/ssl/private
   ```

2. Kopiera certifikat- och nyckelfilerna till `/etc/ssl`-sökvägarna. Exempel:

   ```
   sudo cp tsig-ssl.crt /etc/ssl/certs/

   sudo cp tsig-ssl.key /etc/ssl/private/
   ```

3. (Valfritt) Om du använder ett självsignerat certifikat eller ett PKI-certifikat för SSL/TLS på Tableau Server måste du även kopiera CA-rotcertifikatfilen till datorn med

oberoende gateway. Exempel:

```
sudo cp tableau-server-CA.pem /etc/ssl/certs/
```

Steg 2: Uppdatera miljövariablerna för TLS

Du måste uppdatera port- och protokollmiljövariabler för konfiguration av oberoende gateway.

Ändra de här värdena genom att uppdatera filen `/etc/opt/tableau/tableau_tsig/environment.bash` så här:

```
TSIG_HK_PROTOCOL="https"
TSIG_PORT="443"
TSIG_PROTOCOL="https"
```

Steg 3: Uppdatera stubbkonfigurationsfilen för HK-protokollet

Redigera stubbkonfigurationsfilen manuellt (`/var/opt/tableau/tableau_tsig/config/httpd.conf.stub`) för att ställa in TLS-relaterade Apache httpd-direktiv för HK-protokollet (housekeeping protocol – rensningsprotokoll).

Stubbkonfigurationsfilen innehåller ett block av TLS-relaterade direktiv som kommenteras ut med en `#TLS#`-markör. Ta bort markörerna från direktiven enligt exemplet nedan. Observera att exemplet visar användningen av rot-CA-certifikat för SSL-certifikatet som används på Tableau Server med alternativet `SSLCACertificateFile`.

```
#TLS# SSLPassPhraseDialog exec:/path/to/file
<VirtualHost *:${TSIG_HK_PORT}>
SSLEngine on
#TLS# SSLHonorCipherOrder on
#TLS# SSLCompression off
SSLCertificateFile /etc/ssl/certs/tsig-ssl.crt
SSLCertificateKeyFile /etc/ssl/private/tsig-ssl.key
SSLCACertificateFile /etc/ssl/certs/tableau-server-CA.pem
#TLS# SSLCARevocationFile /path/to/file
</VirtualHost>
```

De här ändringarna går förlorade om du installerar om den oberoende gatewayen. Vi rekommenderar att du gör en säkerhetskopia.

Steg 4: Kopiera stubbfil och starta om tjänsten

1. Kopiera filen som du uppdaterade i det föregående steget för att uppdatera httpd.conf med ändringarna:

   ```
   cp /var/opt/tableau/tableau_tsig/config/httpd.conf.stub
   /var/opt/tableau/tableau_tsig/config/httpd.conf
   ```

2. Starta om oberoende gateway-tjänsten:

   ```
   sudo su - tableau-tsig
   systemctl --user restart tsig-httpd
   exit
   ```

När du har startat om fungerar inte den oberoende gatewayen förrän du kört nästa uppsättning steg på Tableau Server. När du har slutfört stegen på Tableau Server kommer den oberoende gatewayen att anta ändringar och gå online.

Konfigurera Tableau Server-nod 1 för TLS

Kör de här stegen på nod 1 i din Tableau Server-driftsättning.

Steg 1: Kopiera certifikat och nycklar och stoppa TSM

1. Verifiera att du har Tableau Server-certifikaten ”extern SSL” och nycklarna kopierade till kod 1.

2. För att minimera stilleståndstiden rekommenderar vi att du stoppar TSM, kör följande steg och sedan startar TSM efter att ändringar har tillämpats:

   ```
   tsm stop
   ```

Steg 2: Ställ in certifikatresurser och aktivera oberoende gateway-konfiguration

1. Ange platsen för certifikat och nyckelfiler för oberoende gateway. De här sökvägarna hänvisar till platsen på datorerna med oberoende gateway. Observera att det här exemplet förutsätter att samma certifikat och nyckelpar används för att skydda HTTPS och rensningstrafik:

   ```
   tsm configuration set -k gateway.tsig.ssl.cert.file_name -v
   /etc/ssl/certs/tsig-ssl.crt --force-keys
   tsm configuration set -k gateway.tsig.ssl.key.file_name -v
   /etc/ssl/private/tsig-ssl.key --force-keys
   ```

2. Aktivera TLS för HTTPS- och HK-protokoll för oberoende gateway:

   ```
   tsm configuration set -k gateway.tsig.ssl.enabled -v true --
   force-keys
   tsm configuration set -k gateway.tsig.hk.ssl.enabled -v true --
   force-keys
   ```

3. (Valfritt) Om du använder ett självsignerat certifikat eller ett PKI-certifikat för SSL/TLS på den oberoende gatewayen måste du ladda upp CA-rotcertifikatfilen. CA-rotcertifikatfilen är rotcertifikatet som användes för att generera certifikaten för datorerna med oberoende gateway. Exempel:

   ```
   tsm security custom-cert add -c rootTSIG-CACert.pem
   ```

4. (Valfritt) Om du använder ett självsignerat certifikat eller ett PKI-certifikat för SSL/TLS på Tableau Server måste du kopiera CA-rotcertifikatfilen till katalogen för oberoende gateway `/etc/ssl/certs`. CA-rotcertifikatfilen är rotcertifikatet som användes för att generera certifikaten för Tableau Server-datorerna. Efter att du har kopierat certifikatet till den oberoende gatewayen måste du ange platsen för certifikatet på nod 1 med följande tsm-kommando. Exempel:

```
tsm configuration set -k gateway.tsig.ssl.proxy.gateway_relay_
cluster.cacertificatefile -v /etc/ssl/certs/tableau-server-
CA.pem --force-keys
```

5. (Valfritt: endast för teständamål) Om du använder självsignerade certifikat eller PKI-certifikat för delning mellan datorer och ämnesnamnen på certifikaten därför inte matchar datornamnen måste du inaktivera certifikatverifiering.

   ```
   tsm configuration set -k gateway.tsig.ssl.proxy.verify -v
   optional_no_ca --force-keys
   ```

Steg 3: Aktivera ”extern SSL” för Tableau Server och tillämpa ändringar

1. Aktivera och konfigurera ”extern SSL” på Tableau Server:

   ```
   tsm security external-ssl enable --cert-file ts-ssl.crt --key-
   file ts-ssl.key
   ```

2. Tillämpa ändringarna.

   ```
   tsm pending-changes apply
   ```

Steg 4: Uppdatera JSON-filen för gatewaykonfiguration och starta tsm

1. Uppdatera konfigurationsfilen för oberoende gateway (till exempel `tsig.json`) på Tableau Server-sidan för att ange `https`-protokollet för oberoende gateway-objekten:

   ```
   "protocol" : "https",
   ```

2. Ta bort (eller kommentera ut) anslutningsinformationen för den andra instansen av oberoende gateway. Se till att verifiera JSON i en extern redigerare innan du sparar den.

   När du har konfigurerat och validerat TLS för den enskilda instansen av oberoende gateway uppdaterar du den här JSON-filen med anslutningsinformationen för den andra instansen av oberoende gateway.

3. Kör följande kommando för att uppdatera konfigurationen av oberoende gateway:

```
tsm topology external-services gateway update -c tsig.json
```

4. Starta TSM.

   ```
   tsm start
   ```

5. Medan TSM startar loggar du in på instansen för oberoende gateway och startar om`tsig-httpd`-tjänsten:

   ```
   sudo su - tableau-tsig
   systemctl --user restart tsig-httpd
   exit
   ```

Uppdatera IdP-autentiseringsmodulens URL:er till HTTPS

Om du har konfigurerat en extern identitetsleverantör för Tableau måste du sannolikt uppdatera returadresser (URL) i den administrativa instrumentpanelen för IdP.

Om du till exempel använder ett förauktoriseringsprogram för Okta måste du uppdatera programmet för att använda HTTPS-protokollet för mottagaradressen och destinationsadressen.

Konfigurera AWS-belastningsutjämnare för HTTPS

Om du distribuerar med AWS-belastningsutjämnaren enligt den här guiden konfigurerar du om AWS-belastningsutjämnaren så att den skickar HTTPS-trafik till datorerna som kör oberoende gateway:

1. Ta bort den befintliga HTTP-målgruppen:

   I **Målgrupper** väljer du den HTTP-målgrupp som har konfigurerats för belastningsbalansen, klickar på **Åtgärder** och sedan på **Ta bort**.

2. Skapa HTTPS-målgrupp:

   Målgrupper > **Skapa målgrupp**

- Välj ”Instanser”
- Ange ett namn på målgruppen, till exempel `TG-internal-HTTPS`
- Välj din VPC
- Protokoll: HTTPS 443
- Under **Hälsokontroller** > **Avancerade inställningar för hälsokontroller** > **Framgångskoder** lägger du till kodlistan enligt följande: `200,303`.
- Klicka på **Skapa**.

3. Välj den målgrupp som du just skapade och klicka sedan på fliken **Mål**:

   - Klicka på **Redigera**
   - Välj den EC2-instans som kör den oberoende gateway för Tableau Server som du har konfigurerat och klicka sedan på **Lägg till bland registrerade**.
   - Klicka på **Spara**.

4. När målgruppen har skapats måste du aktivera varaktighet:

   - Öppna sidan AWS-målgrupp (**EC2** > **Belastningsutjämning** > **Målgrupper**) och välj den målgruppsinstans som du just konfigurerade. I **Åtgärdsmenyn** väljer du **Redigera attribut**.
   - På sidan **Redigera attribut** väljer du **Varaktighet**, anger en varaktigheten `1 day` och trycker sedan på **Spara ändringar**.

5. Uppdatera lyssnarreglerna vid belastningsutjämning. Välj den belastningsutjämnare du har konfigurerat för den här driftsättningen och klicka sedan på fliken **Lyssnare** .

   - För **HTTP:80** klickar du på **Visa/redigera regler**. Öppna sidan **Regler** och klicka därefter på redigeringsikonen (en gång högst upp på sidan och sedan igen bredvid regeln) för att redigera regeln. Ta bort den befintliga THEN-regeln och ersätt den genom att klicka på **Lägg till åtgärd** > **Omdirigera till ...**. I den resulterande SEDAN-konfigurationen anger du `HTTPS` och port `443` och låter övriga alternativ behålla standardinställningarna. Spara inställningen och klicka sedan på **Uppdatera**.
   - För **HTTPS:443** klickar du på **View/edit rules (Visa/redigera regler)**. Öppna sidan **Regler** och klicka därefter på redigeringsikonen (en gång högst upp på sidan och sedan igen bredvid regeln) för att redigera regeln. Ta bort befintlig THEN-regel och ersätt genom att klicka på **Lägg till åtgärd** > **Vidarebefordra**

till .... Ange målgruppen för HTTPS-gruppen som du just skapade. Aktivera varaktighet under **Varaktighet på gruppnivå** och ange varaktigheten till 1 dag. Spara inställningen och klicka sedan på **Uppdatera**.

6. Uppdatera tidsgränsen för inaktivitet till 400 sekunder för belastningsutjämnaren. Välj den belastningsutjämnare du har konfigurerat för den här driftsättningen och klicka sedan på **Åtgärder** > **Redigera attribut**. Ställ in **tidsgränsen för inaktivitet** på `400` sekunder och klicka sedan på **Spara**.

Validera TLS

Validera att TLS fungerar genom att logga in på Tableau Server med den offentliga URL:en (t.ex., https://tableau.example.com) med det Tableau-administratörskonto som du skapade i början av den här proceduren.

Om TSM inte startar eller om du får andra fel, se Felsöka oberoende gateway för Tableau Server.

Konfigurera den andra instansen av oberoende gateway för SSL

När du har konfigurerat den första instansen av oberoende gateway driftsätter du den andra instansen.

För att driftsätta den andra oberoende gatewayen behöver du följa stegen nedan:

1. På den konfigurerade (första) instansen av oberoende gateway: kopiera följande filer till motsvarande platser på den andra instansen av oberoende gateway:
   - `/etc/ssl/certs/tsig-ssl.crt`
   - `/etc/ssl/private/tsig-ssl.key` (Du måste skapa katalogen `private` på den andra instansen).
   - `/var/opt/tableau/tableau_tsig/config/httpd.conf.stub`
   - `/etc/opt/tableau/tableau_tsig/environment.bash`

2. På nod 1 i Tableau Server-driftsättningen uppdaterar du anslutningsfilen (`tsig.json`) med anslutningsinformationen från den andra oberoende gatewayen.

   Ett exempel på anslutningsfil (`tsig.json`) visas här:

   ```
   {
   "independentGateways": [
    {
      "id": "ip-10-0-1-169.ec2.internal",
      "host": "ip-10-0-1-169.ec2.internal",
      "port": "21319",
      "protocol" : "https",
      "authsecret": "13660-27118-29070-25482-9518-22453"
    },
    {
      "id": "ip-10-0-2-230.ec2.internal",
      "host": "ip-10-0-2-230.ec2.internal",
      "port": "21319",
      "protocol" : "https",
      "authsecret": "9055-27834-16487-27455-30409-7292"
    }]
    }
   ```

3. På nod 1 i Tableau Server-driftsättningen kör du följande kommandon för att uppdatera konfigurationen:

   ```
   tsm stop

   tsm topology external-services gateway update -c tsig.json

   tsm start
   ```

4. På båda instanserna av oberoende gateway: när Tableau Server startar, starta om`tsig-httpd`-processen på båda instanserna av oberoende gateway:

   ```
   sudo su - tableau-tsig

   systemctl --user restart tsig-httpd
   ```

```
exit
```

5. I AWS **EC2 > Målgrupper**: Uppdatera målgruppen för att inkludera EC2-instansen som kör den andra oberoende gateway-instansen.

   Välj den målgrupp som du just skapade och klicka sedan på fliken Mål:

   - Klicka på **Redigera**.
   - Välj EC2-instansen för datorn med den andra oberoende gatewayen och klicka sedan på **Lägg till i registrerade**. Klicka sedan på **Spara**.

Konfigurera SSL för Postgres

Du kan valfritt konfigurera SSL (TLS) för Postgres-anslutningen för den externa lagringsplatsanslutningen på Tableau Server.

För att underlätta certifikathanteringen och driftsättningen, och som en rekommenderad säkerhetsåtgärd, bör du använda certifikat genererade av någon av de stora betrodda certifikatutfärdarna (CA). Du kan också generera självsignerade certifikat eller använda certifikat från en PKI för TLS.

Denna procedur beskriver hur du använder OpenSSL för att generera självsignerade certifikat på Postgres-värden för en RHEL-lik Linux-distribution i exemplet för AWS-referensarkitektur.

När du har genererat och signerat SSL-certifikatet måste du kopiera CA-certifikatet till Tableau-värden.

På värden som kör Postgres:

1. Generera signeringsnyckel för rotcertifikatutfärdare (CA, Certificate Authority):

   ```
   openssl genrsa -out pgsql-rootCAKey.pem 2048
   ```

2. Skapa certifikat för rot-CA:

```
openssl req -x509 -sha256 -new -nodes -key pgsql-rootCAKey.pem
-days 3650 -out pgsql-rootCACert.pem
```

Du uppmanas att ange värden i certifikatfälten. Till exempel:

```
Country Name (2 letter code) [XX]:US
State or Province Name (full name) []:Washington
Locality Name (eg, city) [Default City]:Seattle
Organization Name (eg, company) [Default Company Ltd]:Tableau
Organizational Unit Name (eg, section) []:Operations
Common Name (eg, Postgres server's hostname) []:ip-10-0-1-
189.us-west-1.compute.internal
Email Address []:example@tableau.com
```

3. Skapa certifikatet och relaterad nyckel (`server.csr` och `server.key` i nedanstående exempel) för Postgres-datorn. Certifikatmottagarens namn måste stämma överens med EC2-instansens privata DNS-namn för Postgres-värden. Certifikatmottagarens namn anges med alternativet `-subj` i formatet `"/CN=<private DNS name>"`, till exempel:

   ```
   openssl req -new -nodes -text -out server.csr -keyout
   server.key -subj "/CN=ip-10-0-1-189.us-west-1.compute.internal"
   ```

4. Signera det nya certifikatet med det CA-certifikat som du skapade i steg 2. Följande kommando skapar även certifikatet i `crt`-format:

   ```
   openssl x509 -req -in server.csr -days 3650 -CA pgsql-
   rootCACert.pem -CAkey pgsql-rootCAKey.pem -CAcreateserial -out
   server.crt
   ```

5. Kopiera crt- och nyckelfilerna till Postgres-sökvägen `/var/lib/pgsql/13/data/`:

   ```
   sudo cp server.crt /var/lib/pgsql/13/data/
   ```

   ```
   sudo cp server.key /var/lib/pgsql/13/data/
   ```

6. Byt till rotanvändare:

   ```
   sudo su
   ```

7. Ställ in behörigheter på cer- och nyckelfilerna. Kör följande kommandon:

```
cd /var/lib/pgsql/13/data
chown postgres.postgres server.crt
chown postgres.postgres server.key
chmod 0600 server.crt
chmod 0600 server.key
```

8. Uppdatera pg_haba-konfigurationsfilen, `/var/lib/pgsql/13/data/pg_hba.conf` för att specificera md5-förtroende:

   Ändra de befintliga anslutningssatserna från

   `host all all 10.0.30.0/24 password` och

   `host all all 10.0.31.0/24 password`

   till

   `host all all 10.0.30.0/24 md5` och

   `host all all 10.0.31.0/24 md5`.

9. Uppdatera postgresql-filen, `/var/lib/pgsql/13/data/postgresql.conf`, genom att lägga till följande rad:

   ```
   ssl = on
   ```

10. Avsluta rotanvändarläget:

    ```
    exit
    ```

11. Starta om Postgres:

    ```
    sudo systemctl restart postgresql-13
    ```

Valfritt: Aktivera certifikatförtroendevalidering på Tableau Server för Postgres SSL

Om du följde installationsproceduren i Del 4 – Installera och konfigurera Tableau Server så är Tableau Server konfigurerad med valfri SSL för Postgres-anslutningen. Detta innebär att

konfigurering av SSL på Postgres (enligt beskrivningen ovan) resulterar i en krypterad anslutning.

Om du vill kräva certifikatförtroendevalidering för anslutningen måste du köra följande kommando på Tableau Server för att konfigurera om Postgres-värdanslutningen:

```
tsm topology external-services repository replace-host -f
<filename>.json -c CACert.pem
```

där `<filename>.json` är anslutningsfilen som beskrivs i Konfigurera extern Postgres. Och `CACert.pem` är CA-certifikatfilen för SSL/TLS-certifikatet som används av Postgres.

Valfritt: Verifiera SSL-anslutning

För att verifiera SSL-anslutning måste du:

- Installera Postgres-klienten på Tableau Server-nod 1.
- Kopiera rotcertifikatet som du skapade i föregående procedur till Tableau-värden.
- Anslut till Postgres-servern från nod 1

Installera Postgres-klienten på nod 1

Det här exemplet visar hur man installerar versionen Postgres 13.4. Installera samma version som du kör för den externa lagringsplatsen.

1. Skapa och redigera filen pgdg.repo i sökvägen `/etc/yum.repos.d` på nod 1. Lägg till följande konfigurationsinformation i filen.

   ```
   [pgdg13]
   name=PostgreSQL 13 for RHEL/CentOS 7 - x86_64

   baseurl=https://download.postgresql.org/pub/repos/yum/13/redhat-
   /rhel-7-x86_64
   enabled=1
   gpgcheck=0
   ```

2. Installera Postgres-klienten:

```
sudo yum install postgresql13-13.4-1PGDG.rhel7.x86_64
```

Kopiera rotcertifikat till nod 1

Kopiera CA-certifikatet (`pgsql-rootCACert.pem`) till Tableau-värden:

```
scp ec2-user@<private-DNS-name-of-Postgress-host>:/home/ec2-
user/pgsql-rootCACert.pem /home/ec2-user
```

Ansluta till Postgres-värden via SSL från nod 1

Kör följande kommando från nod 1, ange värdens IP-adress för Postgres-servern och rot-CA-certifikatet:

```
psql "postgresql://postgres@<IP-
address>:5432/postgres?sslmode=verify-ca&sslrootcert=pgsql-
rootCACert.pem"
```

Exempel:

```
 psql
"postgresql://postgres@10.0.1.189:5432/postgres?sslmode=verify-
ca&sslrootcert=pgsql-rootCACert.pem"
```

Postgres ber dig om lösenordet. När du har loggat in visas följande i gränssnittet:

```
psql (13.4)
SSL connection (protocol: TLSv1.2, cipher: ECDHE-RSA-AES256-GCM-
SHA384, bits: 256, compression: off)
Type "help" for help.
postgres=#
```

Konfigurera SMTP- och händelsemeddelanden

Tableau Server skickar e-postmeddelanden till administratörer och användare. För att aktivera detta måste du konfigurera Tableau Server för att skicka e-post till din e-postserver. Du måste också ange de händelsetyper, de tröskelvärden och den prenumerationsinformation som du vill ska skickas.

För den initiala konfigurationen av SMTP och meddelanden rekommenderar vi att du använder konfigurationsfilmallen nedan för att skapa en json-fil. Du kan också ställa in en enskild konfigurationsnyckel som listas nedan med beskriven syntax *tsm-konfigurationsuppsättning* (Linux).

Kör denna procedur på Nod 1 i din Tableau Server-driftsättning:

1. Kopiera följande JSON-mall till en fil. Anpassa filen med dina SMTP-konfigurationsalternativ och prenumerations- och varningsmeddelanden för din organisation.
    - För att se en lista och beskrivning av alla SMTP-alternativ, se *SMTP CLI-konfigurationsreferens* (Linux).
    - För att se en lista och beskrivning av alla alternativ för aviseringshändelser, se CLI-avsnittet i *Konfigurera aviseringar för serverhändelser* (Linux).

```
{
"configKeys": {
       "svcmonitor.notification.smtp.server": "SMTP server host
name",
       "svcmonitor.notification.smtp.send_account": "SMTP user name",
       "svcmonitor.notification.smtp.port": 443,
       "svcmonitor.notification.smtp.password": "SMTP user account
password",
       "svcmonitor.notification.smtp.ssl_enabled": true,
       "svcmonitor.notification.smtp.from_address": "From email
address",
```

```
        "svcmonitor.notification.smtp.target_addresses": "To email
address1,address2",
        "svcmonitor.notification.smtp.canonical_url": "Tableau Server
URL",
        "backgrounder.notifications_enabled": true,
        "subscriptions.enabled": true,
        "subscriptions.attachments_enabled": true,
        "subscriptions.max_attachment_size_megabytes": 150,
        "svcmonitor.notification.smtp.enabled": true,
        "features.DesktopReporting": true,
        "storage.monitoring.email_enabled": true,
        "storage.monitoring.warning_percent": 20,
        "storage.monitoring.critical_percent": 15,
        "storage.monitoring.email_interval_min": 25,
        "storage.monitoring.record_history_enabled": true
        }
}
```

2. Kör `tsm settings import -f file.json` för att skicka json-filen till Tableau Services Manager.
3. Tillämpa ändringarna genom att köra kommandot `tsm pending-changes apply`.
4. Visa och verifiera anslutningskonfigurationen genom att köra kommandot `tsm email test-smtp-connection`.

Installera PostgreSQL-drivrutin

För att visa administratörsvyer på Tableau Server måste PostgreSQL-drivrutinen vara installerad på nod 1 i Tableau Server-driftsättningen.

1. Gå till nedladdningssidan för Tableau-drivrutiner och kopiera webbadressen för jar-filen PostgreSQL.
2. Kör följande procedur på varje enskild nod i Tableau-distributionen:

- Skapa följande filsökväg:

  ```
  sudo mkdir -p /opt/tableau/tableau_driver/jdbc
  ```

- Hämta den senaste versionen av jar-filen PostgreSQL från den nya sökvägen. Till exempel:

  ```
  sudo wget
  https://downloads.tableau.com/drivers/linux/postgresql/pos-
  tgresql-42.2.22.jar
  ```

3. Starta om Tableau Server på den initiala noden:

   ```
   tsm restart
   ```

Konfigurera stark lösenordspolicy

Om du inte distribuerar Tableau Server med en IDP-autentiseringslösning, rekommenderar vi att du säkerhetshärdar den förvalda lösenordspolicyn för Tableau.

Om du distribuerar Tableau Server med en IdP måste du hantera lösenordspolicyer med denna IdP.

Följande procedur inkluderar json-konfiguration för inställning av lösenordspolicy på Tableau Server. För mer information om alternativen nedan, se *Lokal autentisering* (Linux).

1. Kopiera följande JSON-mall till en fil. Fyll i nyckelvärden med konfigurationen för lösenordspolicy.

   ```
   {
    "configKeys": {
          "wgserver.localauth.policies.mustcontainletters.enabled":
   true,
          "wgserver.localauth.policies.mustcontainuppercase.enabled":
   true,
          "wgserver.localauth.policies.mustcontainnumbers.enabled":
   ```

```
true,
       "wgserver.localauth.policies.mustcontainsymbols.enabled":
true,
       "wgserver.localauth.policies.minimumpasswordlength.enabled":
true,
       "wgserver.localauth.policies.minimumpasswordlength.value": 12,
       "wgserver.localauth.policies.maximumpasswordlength.enabled":
false,
       "wgserver.localauth.policies.maximumpasswordlength.value":
255,
       "wgserver.localauth.passwordexpiration.enabled": true,
       "wgserver.localauth.passwordexpiration.days": 90,
       "wgserver.localauth.ratelimiting.maxbackoff.minutes": 60,
       "wgserver.localauth.ratelimiting.maxattempts.enabled": false,
       "wgserver.localauth.ratelimiting.maxattempts.value": 5,
       "vizportal.password_reset": true
 }
}
```

2. Kör `tsm settings import -f file.json` för att skicka json-filen till Tableau Services Manager i syfte att konfigurera Tableau Server.

3. Tillämpa ändringarna genom att köra kommandot `tsm pending-changes apply`.

Del 7 – Validering, verktyg och felsökning

Den här delen inkluderar valideringssteg och felsökningsvägledning efter installation.

Validering av reservomkopplingssystem

När du har konfigurerat din driftsättning rekommenderar vi att du kör enkla reservomkopplingstest för att validera systemets redundans.

Vi rekommenderar att du kör följande steg för att validera att reservomkopplingen fungerar:

1. Stäng av den första instansen av oberoende gateway (TSIG1). All inkommande trafik ska gå genom den andra instansen av oberoende gateway (TSIG2).
2. Starta om TSIG1 och stäng sedan av TSIG2. All inkommande trafik ska gå genom TSIG1.
3. Starta om TSIG2.

4. Stäng av Tableau Server-nod 1. All servicetrafik för Vizportal/Program reservomkopplas till nod 2.

   Obs! Från och med september 2022 innebär hög tillgänglighet för nod 1 en säkerhetsrisk i vissa versioner av Tableau Server 2021.4 och senare. Klientanslutningar misslyckas om nod 1 ligger nere. Det här problemet har korrigerats i dessa underhållsversioner:

   – 2021.4.15 och senare
   – 2022.1.11 och senare
   – 2023.1.3 och senare

   För att försäkra dig om att er Tableau Server-installation med ATR-aktiveringar har

> en tidsfrist på 72 timmar vid fel på den initiala noden installerar du eller uppgraderar till en av dessa versioner. Mer information finns i Tableau Server HA using ATR Does Not Have a Grace Period After the Initial Node Failure (på engelska) i Tableaus kunskapsbas.

5. Starta om nod 1 och stäng av nod 2. All servicetrafik för Vizportal/Program reservomkopplas till nod 1.
6. Starta om nod 2.

I det här sammanhanget görs ”avstängning” eller ”omstart” genom att stänga av operativsystemet eller den virtuella datorn utan att först försöka stänga av programmet. Målet är att simulera ett fel på maskinvaran eller den virtuella datorn.

Det minsta valideringssteget för varje reservomkopplingstest är att autentisera med en användare och utföra grundläggande vyaktiviteter.

Du kan få webbläsarfelet ”Felaktig begäran” när du försöker logga in efter ett simulerat fel. Du kan se det här felet även om du rensar cacheminnet i webbläsaren. Det här problemet uppstår ofta när webbläsaren cachelagrar data från tidigare IdP-sessioner. Om felet kvarstår även efter att du rensat den lokala webbläsarens cacheminne så kan du validera Tableau-scenariot genom att ansluta med en annan webbläsare.

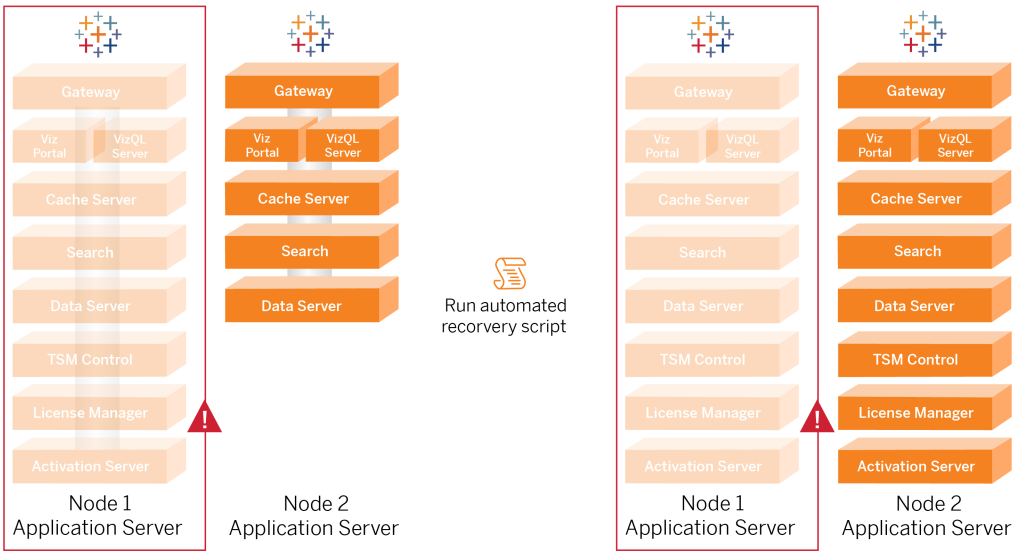
Automatisk återställning av ursprunglig nod

Tableau Server version 2021.2.4 och senare har ett skript för automatisk återställning av ursprunglig nod, `auto-node-recovery`, i katalogen scripts (`/app/tableau_server/packages/scripts.<version>`).

Om det finns ett problem med den ursprungliga noden och du har redundanta processer på nod 2 så finns det ingen garanti att Tableau Server kommer fortsätta att fungera. Tableau Server kan fortsätta att fungera i upp till 72 timmar efter ett fel på den ursprungliga noden innan bristen på licensieringsserver börjar påverka andra processer. I sådant fall kan det hända att användarna kan fortsätta att logga in samt se och använda sitt innehåll efter det att

den ursprungliga noden misslyckas, men du kommer inte att kunna omkonfigurera Tableau Server eftersom du inte kommer att ha åtkomst till administrationsstyrenheten.

Även när Tableau Server har konfigurerats med redundanta processer är det möjligt att den inte fortsätter att fungera efter att den ursprungliga noden misslyckas



För att återställa fel på ursprunglig nod (nod 1):

1. Logga in på Tableau Server nod 2.
2. Ändra till katalogen scripts:

   ```
   cd /app/tableau_server/packages/scripts.<version>
   ```

3. Kör följande kommando för att starta skriptet:

   ```
   sudo ./auto-node-recovery -p node1 -n node2 -k <license keys>
   ```

   Där `<license keys>` är en kommaavgränsad lista (utan blanksteg) med licensnycklarna för din driftsättning. Om du inte har tillgång till dina licensnycklar kan du gå till Tableau-kundportalen och hämta dem. Exempel:

```
sudo ./auto-node-recovery -p node1 -n node2 -k TSB4-8675-309F-
TW50-9RUS,TSNM-559N-ULL6-22VE-SIEN
```

Den automatiska nodåterställningsskriptet kör ungefär 20 steg för att återställa tjänsterna till nod 2. Varje steg visas i terminalen allteftersom skriptet fortskrider. Mer detaljerad status loggas till `/data/tableau_data/logs/app-controller-move.log`. I de flesta miljöer tar skriptet mellan 35 och 45 minuter att slutföra.

Felsöka återställning av ursprunglig nod

Om nodåterställning misslyckas kan du ha hjälp av att köra skriptet interaktivt för att tillåta eller neka diskreta steg i processen. Om skriptet till exempel misslyckas halvvägs igenom processen kan du granska loggfilen, ändra i konfigurationen och därefter köra skriptet igen. Genom att köra i interaktivt läge kan du därefter hoppa över alla steg tills du kommer till det steg som misslyckades.

Kör i interaktivt läge genom att lägga till växeln `-i` i skriptargumentet.

Återskapa den misslyckade noden

Efter att du kört skriptet kommer nod 2 att köra alla tjänster som tidigare fanns på den misslyckade nod 1-värden. Om du vill lägga till i 4-noden måste du distribuera en aktuell Tableau Server-värd med startfilen och konfigurera den som du gjorde för den ursprungliga nod 2 som det anges i del 4. Se Konfigurera nod 2.

switchto

Switchto är ett skript från Tim som gör det enkelt att växla mellan fönster.

1. Kopiera följande fil till en fil som heter `switchto` i startkatalogen på din bastion-värd.

   ```
   #!/bin/bash
   #-----------------------------------------------------------------
   -----
   ```

```
# switchto
#
# Helper function to simplify SSH into the various AWS hosts
when
# following the Tableau Server Enterprise Deployment Guide
(EDG).
#
# Place this file on your bastion host and provide your AWS
hosts'
# internal ip addresses or machine names here.
# Example: readonly NODE1="10.0.3.187"
#
readonly NODE1=""
readonly NODE2=""
readonly NODE3=""
readonly NODE4=""
readonly PGSQL=""
readonly PROXY1=""
readonly PROXY2=""

usage() {
echo "Usage: switchto.sh [ node1 | node2 | node3 | node4 |
pgsql | proxy1 | proxy2 ]"
}


ip=""

case $1 in
       node1)
               ip="$NODE1"
               ;;
       node2)
               ip="$NODE2"
               ;;
```

```
        node3)
                ip="$NODE3"
                ;;
        node4)
                ip="$NODE4"
                ;;
        pgsql)
                ip="$PGSQL"
                ;;
        proxy1)
                ip="$PROXY1"
                ;;
        proxy2)
                ip="$PROXY2"
                ;;
        ?)
                usage
                exit 0
                ;;
        *)
                echo "Unkown option $1."
                usage
                exit 1
                ;;
esac

if [[ -z $ip ]]; then
echo "You must first edit this file to provide the ip addresses
of your AWS hosts."
exit 1
fi

ssh -A ec2-user@$ip
```

2. Uppdatera IP-adresserna i skriptet så de mappar till dina EC2-instanser och spara filen.

3. Tillämpa behörigheter för skriptfilen:

```
sudo chmod +x switchto
```

Syntax:

Växla till en värd genom att köra följande kommando:

```
./switchto <target>
```

Om du till exempel vill växla till nod 1 kör du följande kommando:

```
./switchto node1
```

Felsöka oberoende gateway för Tableau Server

Det kan uppstå många fel när oberoende gateway, Okta, Mellon och SAML på Tableau Server konfigureras. Den vanligaste grundorsaken till fel är ett strängfel. ett snedstreck (`/`) i Okta-URL:n som anges under konfigurationen kan till exempel orsaka ett matchningsfel relaterat till ett SAML-påstående. Det här är bara ett exempel. Det finns många möjligheter att mata in en felaktig sträng i någon av applikationerna under konfigurationen.

Starta om tableau-tsig-tjänsten

Börja (och avsluta) alltid felsökningen genom att starta om tableau-tsig-tjänsten på datorerna med oberoende gateway. Det går snabbt att starta om den här tjänsten och det får ofta den uppdaterade konfigurationen att läsas in från Tableau-servern.

Kör följande kommandon på datorn med oberoende gateway:

```
sudo su - tableau-tsig

systemctl --user restart tsig-httpd

exit
```

Hitta felaktiga strängar

Om du har gjort ett strängfel (kopierat/klistrat in felaktigt, trunkerat en sträng osv.) bör du ta dig tid att gå igenom var och en av inställningarna du har konfigurerat:

- Konfiguration av Okta-förautentisering. Granska noggrant URL:erna som du har angett. Leta efter snedstreck. Verifiera HTTP kontra HTTPS.
- Kommandotolkshistorik för SAML-konfiguration på nod 1. Granska kommandot `tsm authentication saml configure` som du körde. Kontrollera att alla URL:er matchar dem som du har konfigurerat i Okta. Medan du granskar kommandotolkshistoriken på nod 1 ska du verifiera att kommandona `tsm configuration set` som anger Mellon-konfigurationsfilens sökvägar mappar exakt till filsökvägarna där du kopierade filerna på oberoende gateway.
- Mellon-konfiguration på oberoende gateway. Granska kommandotolkshistoriken för att verifiera att du skapade metadata med samma URL-sträng som du har konfigurerat i Okta och Tableau SAML. Kontrollera att alla sökvägar som anges i `/etc/mellon/conf.d/global.conf` är korrekta och att `MellonCookieDomain` är inställd på din rotdomän, inte din Tableau-underdomän.

Söka i relevanta loggar

Om alla strängar verkar vara korrekt inställda bör du granska loggar efter fel.

Tableau Server loggar fel och händelser till dussintals olika loggfiler. Den oberoende gatewayen loggar också till en uppsättning lokala filer. Vi rekommenderar att du granskar dessa loggar i följande ordning.

Loggfiler från oberoende gateway

Standardplatsen för loggfilerna från oberoende gateway är `/var/opt/tableau/tableau_tsig/logs`.

- `access.log`: Den här loggen är användbar eftersom den har poster som visar anslutningar från Tableau Server-noderna. Om du får gateway-fel (startar inte) när du försöker starta TSM och det inte finns några poster i filen `access.log`, så finns det ett kärnanslutningsproblem. Verifiera alltid AWS-säkerhetsgruppens konfiguration som ett

första steg. Ett annat vanligt problem är ett stavfel i `tsig.json`. Om du uppdaterar något i tsig.json bör du köra `tsm stop` innan du kör `tsm topology external-services gateway update -c tsig.json`. Efter att tsig.json har uppdaterats kör du `tsm start`.

- `error.log`: Utöver andra poster innehåller den här loggen SAML- och Mellon-fel.

Tableau Server tabadminagent-loggfil

Uppsättningen av `tabadminagent`-filer (inte `tabadmincontroller`-filer) är de enda relevanta loggfilerna för felsökning av fel relaterade till oberoende gateway.

Du måste hitta var oberoende gateway-fel har loggats till `tabdminagent`. Felen kan finnas på vilken nod som helst, men de finns bara på en nod. Utför följande steg på varje nod i Tableau Server-klustret tills du hittar den ”oberoende” strängen:

1. Leta reda på tabadminagent-loggfilens plats på Tableau Server-noderna 1–4 i EDG-installationen:

   ```
   cd /data/tableau_data/data/tabsvc/logs/tabadminagent
   ```

2. Öppna senaste loggen för att läsa:

   ```
   less tabadminagent_nodeN.log
   ```

   (ersätt N med nodnummer)

3. Sök efter alla instanser av ”Oberoende” och ”oberoende” genom att använda följande sökträng:

   ```
   /ndependent
   ```

   Om det inte finns några matchningar går du till nästa nod och upprepar steg 1–3.

4. När du får en matchning trycker du på `Shift + G` för att hoppa längst ner för att få senaste felmeddelanden.

Läsa in httpd-stubbfilen på nytt

Oberoende gateway hanterar konfigurationen av httpd för Apache. En generisk åtgärd som ofta åtgärdar övergående problem är att läsa in httpd-stubbfilen som seedar den underliggande Apache-konfigurationen på nytt. Kör följande kommandon på båda instanserna av oberoende gateway.

1. Kopiera stubbfilen till httpd.conf:

   ```
   cp /var/opt/tableau/tableau_tsig/config/httpd.conf.stub
   /var/opt/tableau/tableau_tsig/config/httpd.conf
   ```

2. Starta om oberoende gateway-tjänsten:

   ```
   sudo su - tableau-tsig
   systemctl --user restart tsig-httpd
   exit
   ```

Ta bort eller flytta loggfiler

Oberoende gateway loggar alla åtkomsthändelser. Du behöver hantera loggfillagringen för att undvika att fylla upp diskutrymme. Om din disk fylls kan den oberoende gatewayen inte skriva åtkomsthändelser och tjänsten kommer att misslyckas. Följande meddelande loggas till error.log på den oberoende gatewayen:

```
(28)No space left on device: [client 10.0.2.209:54332] AH00646:
Error writing to /var/opt/tableau/tableau_
tsig/logs/access.%Y_%m_%d_%H_%M_%S.log
```

Det här felet resulterar i statusen `DEGRADED` för noden `external` när du kör `tsm status -v` på Tableau-nod 1. Noden `external` i den returnerade statusen hänvisar till oberoende gateway.

För att lösa problemet behöver du ta bort eller flytta access.log-filerna från disken. Loggfiler lagras under `/var/opt/tableau/tableau_tsig/logs`. När du har rensat disken ska du starta om tableau-tsig-tjänsten.

Webbläsarfel

Dålig begäran: Ett vanligt fel för det här scenariot är ett Dålig begäran-fel från Okta. Det här problemet uppstår ofta när webbläsaren cachelagrar data från tidigare Okta-sessioner. Om du till exempel hanterar Okta-applikationer som en Okta-administratör och därefter försöker få åtkomst till Tableau med ett annat Okta-aktiverat konto så kan sessionsdata från administratörsdata orsaka Dålig begäran-felet. Om felet kvarstår även efter att du rensat den lokala webbläsarens cacheminne så kan du testa att validera Tableau-scenariot genom att ansluta med en annan webbläsare.

En annan orsak till felet ”Felaktig begäran” är ett stavfel i en av de många URL:er som du anger under Okta-, Mellon- och SAML-konfigurationsprocesserna. Kontrollera att du har angett alla dessa utan fel.

Ofta kommer filen `error.log` på oberoende gateway-servern att ange vilken URL som orsakar felet.

Hittades inte – Den begärda URL:en hittades inte på den här servern: Det här felet indikerar ett av flera konfigurationsfel.

Om användaren autentiserats med Okta och därefter stöter på det här felet så har du sannolikt laddat upp Okta förautentiseringsapplikationen till Tableau Server när du konfigurerade SAML. Verifiera att du har konfigurerat Okta-applikationsmetadata för Tableau Server på Tableau Server och inte applikationsmetadata för Okta förautentisering.

Andra felsökningssteg:

- Granska applikationsinställningarna för Okta förautentisering. Se till att HTTP- kontra HTTPS-protokollen angetts som specificerade i det här ämnet.
- Starta om tsig-httpd på båda oberoende gateway-servrarna.
- Verifiera att `sudo apachectl configtest` returnerar ”Syntax OK” för bägge oberoende gateways.
- Verifiera att testanvändaren tilldelats båda applikationerna i Okta.
- Verifiera att ihållande angetts för lastbalanseraren och associerade målgrupper.

Verifiera TLS-anslutning från Tableau Server till oberoende gateway

Använd kommandot `wget` för att verifiera anslutning och åtkomst från Tableau Server till oberoende gateway. Variationer av det här kommandot kan hjälpa dig att förstå om certifikatproblem orsakar anslutningsproblem.

Kör till exempel det här `wget`-kommando för att verifiera rensningsprotokollet (HK) från Tableau Server:

```
wget https://ip-10-0-1-38.us-west-1.compute.internal:21319
```

Konstruera URL:en med samma värdadress som du inkluderade för värdalternativet för filen tsig.json. Specificera `https`-protokollet och lägg till URL:en med HK-porten `21319`.

Så här kontrollerar du anslutningen och ignorerar certifikatverifiering:

```
wget https://ip-10-0-1-38.us-west-1.compute.internal:21319 --no-check-certificate
```

Så här verifierar du att rot-CA-certifikatet för TSIG är giltigt:

```
wget https://ip-10-0-1-38.us-west-1.compute.internal:21319 --ca-certificate=tsigRootCA.pem
```

Om Tableau kan kommunicera kan du fortfarande få innehållsrelaterade fel, men du kommer inte att få anslutningsrelaterade fel. Om Tableau inte kan ansluta alls ska du börja med att verifiera protokollkonfigurationen i brandväggen/säkerhetsgrupperna. Till exempel måste reglerna för inkommande trafik för säkerhetsgruppen där oberoende gateway finns tillåta TCP 21319.

Bilaga – AWS Deployment Toolbox

Det här ämnet erbjuder verktyg och alternativa driftsättningsalternativ för referensarkitekturen när den driftsätts i AWS. Mer bestämt beskrivs i det här avsnittet hur du automatiserar det exempel på AWS-driftsättning som används i hela EDG.

TabDeploy4EDG – automatiserat installationsskript

TabDeploy4EDG-skriptet automatiserar implementeringen av Tableau-driftsättningen med fyra noder som beskrivs i Del 4 – Installera och konfigurera Tableau Server. Om du följer exemplet på AWS-implementeringen som beskrivs i den här guiden kan du kanske köra TabDeploy4EDG.

Krav. För att köra skriptet måste du först förbereda och konfigurera AWS-miljön enligt exemplet med implementeringen i Del 3 – förbereda för företagsdriftsättning av Tableau Server:

- VPC-, subnät- och säkerhetsgrupper har konfigurerats enligt beskrivningen. IP-adresser behöver inte matcha de som visas i exemplet med implementeringen.
- Fyra EC2-instanser som kör de senaste och uppdaterade versionerna av AWS Linux 2
- PostgreSQL är installerat och har konfigurerats såsom beskrivs i Installera, konfigurera och tar PostgreSQL.
- Tar-backupfilen för steg 1 finns på EC2-instansen där PostgreSQL är installerat, såsom beskrivs i Skapa tar-säkerhetskopia (PostgreSQL steg 1).
- EC2-instansen som ska köra nod 1 av Tableau Server-driftsättning har konfigurerats för att kommunicera med PostgreSQL såsom beskrivs i Del 4 – Installera och konfigurera Tableau Server.
- Du har loggat in på varje EC2-instans med en SSH-session från bastionvärden.

Skriptet tar cirka 1,5–2 timmar att installera och konfigurera de fyra Tableau-servrarna. Skriptet konfigurerar Tableau enligt de föreskrivna inställningarna från referensarkitekturen. Skriptet utför följande åtgärder:

- Återställer en steg 1-säkerhetskopia av PostgreSQL-värden om du anger en sökväg till PostgreSQL-värdens tar-fil.
- Tar bort befintliga Tableau-installationer på alla noder.
- Kör `sudo yum update` på alla noder.
- Laddar ner och kopierar Tableau rpm-paketet till varje nod.
- Laddar ner och installerar beroenden till varje nod.
- Skapar `/app/tableau_server` och installerar paket på alla noder.
- Installerar nod 1 med ett lokalt identitetsregister och konfigurerar en extern lagringsplats med PostgreSQL.
- Utför en bootstrap-installation och initial konfiguration av nod 2 – nod 4.
- Tar bort bootstrap-filen och konfigurationsfilen för TabDeploy4EDG.
- Konfigurerar tjänster över Tableau-klustret enligt specifikationerna i referensarkitekturen.
- Validerar installationen och returnerar status för varje nod.

Ladda ner och kopiera skriptet till bastionvärden

1. Kopiera skriptet från exempelsidan med TabDeploy4EDG och klistra in koden i en fil som heter `TabDeploy4EDG`.
2. Spara filen i arbetskatalogen på EC2-värden som fungerar som bastionvärd.
3. Kör följande kommando för att ändra läget på filen och göra den körbar:

```
sudo chmod +x TabDeploy4EDG
```

Köra TabDeploy4EDG

TabDeploy4EDG måste köras från bastionvärden. Skriptet antar att du kör under sammanhanget ssh forward agent såsom beskrivs i Exempel: Anslut till en bastion-värd i AWS. Om du inte kör med sammanhanget ssh forward agent blir du tillfrågad om lösenord under hela installationsprocessen.

1. Skapa, redigera och spara en registreringsfil (`registration.json`). Filen måste vara en korrekt formaterad json-fil. Kopiera och anpassa följande mall:

```
{
        "zip" : "97403",
```

```
        "country" : "USA",
        "city" : "Springfield",
        "last_name" : "Simpson",
        "industry" : "Energy",
        "eula" : "yes",
        "title" : "Safety Inspection Engineer",
        "phone" : "5558675309",
        "company" : "Example",
        "state" : "OR",
        "department" : "Engineering",
        "first_name" : "Homer",
        "email" : "homer@example.com"
}
```

2. Kör följande kommando för att skapa en konfigurationsfil som mall:

   ```
   ./TabDeploy4EDG -g edg.config
   ```

3. Öppna konfigurationsfilen för att redigera:

   ```
   sudo nano edg.config
   ```

   Minsta kravet är att du lägger till IP-adresserna för varje EC2-värd, en sökväg till registreringsfilen och en giltig licensnyckel.

4. Spara konfigurationsfilen och stäng den sedan när du är klar med att redigera den.

5. Kör följande kommando för att köra TabDeploy4EDG:

   ```
   ./TabDeploy4EDG -f edg.config
   ```

Exempel: Automatisera driftsättning av AWS-infrastruktur med Terraform

Det här avsnittet beskriver hur du konfigurerar och kör Terraform för att driftsätta EDG-referensarkitekturen i AWS. Exemplet på Terraform-konfigurationen som presenteras här

driftsätter en AWS VPC med subnäten, säkerhetsgrupperna och EC2-instanserna som beskrivs i Del 3 – förbereda för företagsdriftsättning av Tableau Server.

Exempel på Terraform-mallar finns på Tableau-exempelwebbplatsen på https://help.tableau.com/samples/en-us/edg/edg-terraform.zip . Dessa mallar måste konfigureras och anpassas för varje enskild organisation. Konfigurationsinnehållet i det här avsnittet beskriver de minsta nödvändiga ändringarna i mallarna som du måste anpassas för driftsättning.

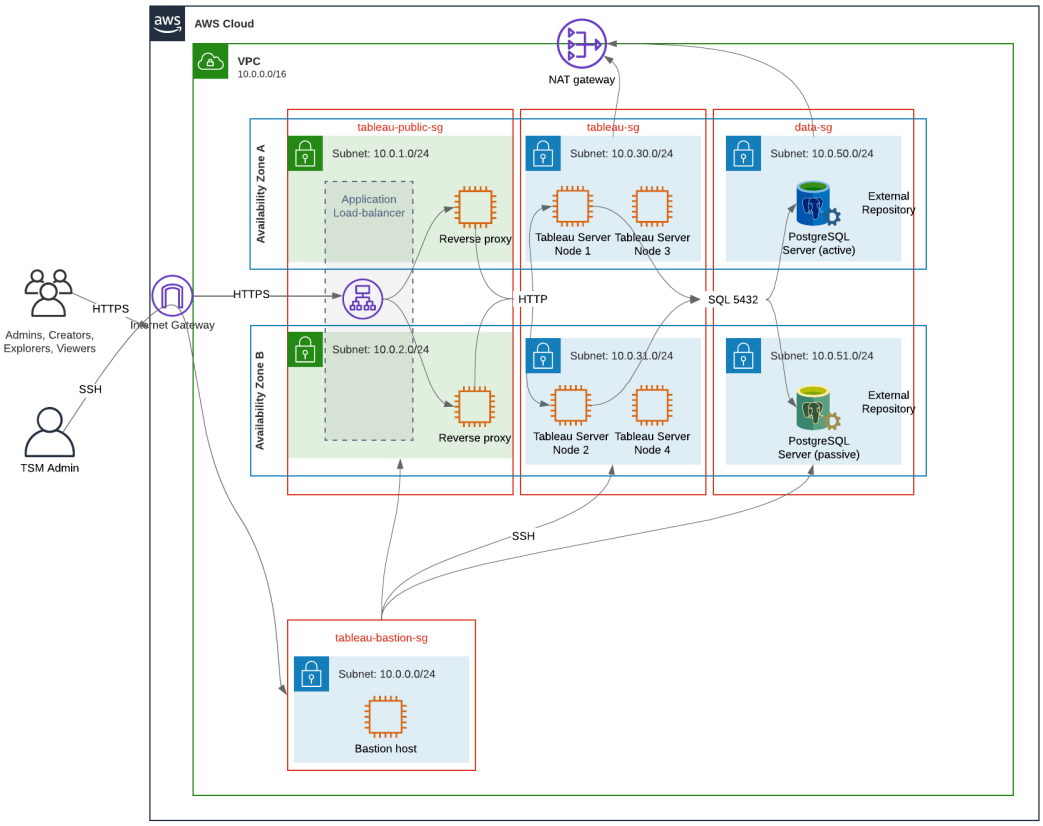
Mål

Terraform-mallarna och innehållet som tillhandahålls här är avsedda att erbjuda ett fungerande exempel som låter dig snabbt driftsätta EDG i en utvecklingstestmiljö.

Vi har gjort vårt bästa för att verifiera och dokumentera exemplet på Terraform-driftsättningen. Att använda Terraform för att driftsätta och underhålla EDG i en produktionsmiljö kräver dock Terraform-expertis som inte inkluderas i det här exemplet. Tableau erbjuder inte support för exemplet på Terraform-lösningen som dokumenteras här.

Sluttillstånd

Följ proceduren i det här avsnittet för att konfigurera en VPC i AWS som är funktionellt likvärdig med den VPC som anges i Del 3 – förbereda för företagsdriftsättning av Tableau Server.



Exempel på Terraform-mallar och stödjande innehåll i det här avsnittet:

- Skapar en VPC med en elastisk IP-adress, två tillgänglighetszoner och organisation av subnät såsom visas ovan (IP-adresserna kan skilja sig)
- Skapar säkerhetsgrupperna Bastion, Offentlig, Privat och Data.
- Konfigurerar de flesta in- och utträdesregler för säkerhetsgrupperna. Säkerhetsgrupperna måste redigeras efter att Terraform har körts.
- Skapar följande EC2-värdar (var och en kör AWS Linux2): bastion, proxy 1 proxy 2, Tableau nod 1, Tableau nod 2, Tableau nod 3, Tableau nod 4.
- EC2-värdar för PostgreSQL skapas inte. EC2 måste skapas manuellt i säkerhetsgruppen Data. PostgreSQL ska sedan installeras och konfigurera enligt beskrivningen i Installera, konfigurera och tar PostgreSQL.

Krav

- AWS-konto – du måste ha tillgång till ett AWS-konto som gör det möjligt att skapa VPC:er.
- Om du kör Terraform från en Windows-dator måste AWS CLI installeras.
- En tillgänglig elastisk IP-adress för AWS-kontot.
- En domän som är registrerad i AWS Route 53. Terraform skapar en DNS-zon och relaterade SSL-certifikat i Route 53. Därför måste profilen som används för att köra Terraform även ha lämpliga behörigheter i Route 53.

Innan du börjar

- Exemplen på kommandoraderna i den här proceduren är för Terminal med Apple OS. Om Terraform körs på Windows kan kommandon behöva anpassas med filsökvägar, efter behov.
- Ett Terraform-projekt består av många textbaserade konfigurationsfiler (.tf-filtillägg). Terraform kan konfigureras genom att anpassa dessa filer. Atom eller Text++ kan användas då de är robusta textredigerare.
- Om Terraform-projektet delas med andra rekommenderar vi att projektet sparas i Git för förändringshantering.

Steg 1 – förbereda miljön

A. Ladda ner och installera Terraform:

https://www.terraform.io/downloads

B. Generera privata-offentliga nyckelpar

Detta är nyckeln som används för att komma åt AWS och den resulterande VPC-miljön. När Terraform körs inkluderas även den publika nyckeln.

Öppna Terminal och kör följande kommando:

1. Create a private key. For example, `my-key.pem`:

   ```
   openssl genrsa -out my-key.pem 1024
   ```

2. Skapa en offentlig nyckel. Detta nyckelformat används inte för Terraform. Det konverteras till en ssh-nyckel vid ett senare tillfälle i den här proceduren:

   ```
   openssl rsa -in my-key.pem -pubout > my-key.pub
   ```

3. Ange behörigheter för den privata nyckeln:

   ```
   sudo chmod 0600 my-key.pem
   ```

   Så ställer du in behörigheter i Windows:

   - Leta upp filen i Utforskaren, högerklicka på den och välj **Egenskaper**. Gå till fliken **Säkerhet** och klicka på **Avancerat**.
   - Ändra ägare till dig själv, inaktivera arv och ta bort alla behörigheter. Ge dig själv **full kontroll** och klicka sedan på **Spara**. Markera filen som skrivskyddad.

4. Skapa en offentlig ssh-nyckel. Det här är nyckeln som ska kopieras till Terraform senare i processen.

   ```
   ssh-keygen -y -f my-key.pem >my-key-ssh.pub
   ```

C. Ladda ner projektet och lägg till en tillståndskatalog

1. Ladda ner och packa upp EDG Terraform-projektet och spara det på den lokala datorn. När nedladdningen har packats upp finns en katalog på toppnivå som heter edg-terraform samt en serie underkataloger.

2. Skapa en katalog som heter `state` , som är likvärdig katalogen `edg-terraform` som finns på toppnivå.

Steg 2: Anpassa Terraform-mallarna

Terraform-mallarna måste anpassas för att passa till AWS- och EDG-miljöerna. Exemplet som följer visar de minsta passningar av mallarna som de flesta organisationer behöver göra. Det är troligt att just din miljö kan kräva andra anpassningar.

Det här avsnittet är organiserat per mallnamn.

Se till att spara alla ändringar innan du fortsätter till *steg 3 – köra Terraform* .

versions.tf

There are three instances of `versions.tf` files where the `required_version` field must match the version of terraform.exe you're using. Check the version of terraform (`terraform.exe -version`) and update each of the following instances:

- `edg-terraform\versions.tf`
- `edg-terraform\modules\proxy\versions.tf`
- `edg-terraform\modules\tableau_instance\versions.tf`

key-pair.tf

1. Öppna den publika nyckeln som genererades i steg 1B och kopiera den:

   ```
   less my-key-ssh.pub
   ```

   Windows: Kopiera innehållet i den publika nyckeln.

2. Kopiera den publika nyckelsträngen till argumentet public_key, såsom:

   ```
   resource "aws_key_pair" "tableau" {
   key_name = "my-key"
   public_key = "ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAAAgQ (truncated
   example) dZVHambOCw=="
   ```

Ensure that the `key_name` value is unique in the datacenter or `terraform apply` will fail.

locals.tf

Update `user.owner` to your name or alias. The value you enter here will be used for the "Name" tag in AWS on the resources that Terraform creates.

providers.tf

1. Lägg till taggar enligt organisationens krav. Exempel:

   ```
   default_tags {
     tags = {
   ```

```
    "Application" = "tableau",
    "Creator" = "alias@example.com",
    "DeptCode" = "8675309",
    "Description" = "EDG",
    "Environment" = "test",
    "Group" = "itcloud@example.com"
   }
}
```

2. If using `provider`, comment out the `assume_role` lines:

```
/* assume_role {
role_arn     = "arn:aws:iam::310946706895:role/terraform-
backend"
session_name = "terraform"
}*/
```

elb.tf

Under `'resource "aws_lb" "tableau" {'` choose a unique value to use for `name` and `tags.Name`.

If another AWS load balancer has the same name in the datacenter, then `terraform apply` will fail.

Add `idle_timeout`:

```
resource "aws_lb" "tableau" {
name                       = "edg-again-alb"
load_balancer_type         = "application"
subnets                    = [for subnet in aws_subnet.public :
subnet.id]
security_groups            = [aws_security_group.public.id]
drop_invalid_header_fields = true
idle_timeout               = 400
tags = {
```

```
Name = "edg-again-alb"
}
                        }
```

variables.tf

Uppdatera rotdomännamnet. Det här namnet måste matcha domänen som du har registrerat i Route 53.

```
variable "root_domain_name" {
 default = "example.com"
 }
```

Som standard specificeras underdomänen, `tableau` för VPC DNS-domännamnet. Uppdatera `subdomain` för att ändra detta:

```
variable "subdomain" {
 default = "tableau"
}
```

modules/tableau_instance/ec2.tf

There are two ec2.tf files in the project. This customization is for the Tableau instance of the ec2.tf in the directory: `modules/tableau_instance/ec2.tf`.

- Lägg till taggar blob om det behövs:

```
tags = {
 "Name" : var.ec2_name,
 "user.owner" = "ALIAS",
 "Application" = "tableau",
 "Creator" = "ALIAS@example.com",
 "DeptCode" = "8675309",
 "Description" = "EDG",
 "Environment" = "test",
 "Group" = "itcloud@example.com"
 }
}
```

- Vid behov kan du eventuellt uppdatera lagringen för att hantera datakrav:

  Rotvolym:

  ```
  root_block_device {
   volume_size = 100
   volume_type = "gp3"
  }
  ```

  Applikationsvolym:

  ```
  resource "aws_ebs_volume" "tableau" {
   availability_zone = data.aws_subnet.tableau.availability_zone
   size              = 500
   type              = "gp3"
  }
  ```

Steg 3 - kör Terraform

A. Initiera Terraform

Byt till katalogen `edg-terraform` i Terminal och kör följande kommando:

```
terraform init
```

Fortsätt till nästa steg om initieringen lyckas. Följ instruktionerna i Terraform-utmatningen om initieringen misslyckades.

B. Planera Terraform

Kör planneringskommandot från samma katalog:

```
terraform plan
```

Det här kommandot kan köras flera gånger. Kör så många gånger som behövs för att åtgärda fel. Fortsätt till nästa steg när det här kommando körs felfritt.

C. Tillämpa Terraform

Kör tillämpningskommandot från samma katalog:

```
terraform apply
```

Terraform will prompt you to verify deployment, type `Yes`.

Valfritt: Förstör Terraform

Du kan förstöra hela VPC:n genom att köra förstörelsekommandot:

```
terraform destroy
```

Förstörelsekommandot kommer bara att förstöra det som har skapats. Om manuella ändringar har gjorts av vissa objekt i AWS (såsom säkerhetsgrupper, subnät, osv.), misslyckas `destroy`. Ange `Control + C`. VPC:n måste sedan rensas upp manuellt till det tillstånd där den var när Terraform ursprungligen skapade den. Du kan sedan köra kommandot `destroy`.

Steg 4 - ansluta till Bastion

Alla kommandoradsanslutningar sker via bastionvärden på TCP 22 (SSH-protokoll).

1. Skapa en inkommande regel i AWS i säkerhetsgruppen Bastion (**AWS** > **Säkerhetsgrupper** > **Bastion SG** > **Redigera inkommande regler**) och skapa en regel för att tillåta SSH (TCP 22)-anslutningar från IP-adressen eller subnätmasken där Terminal-kommandon kommer att köras.

   Valfritt: Det kan vara bra att tillåta filkopiering mellan EC2-instanserna i de privata och offentliga grupperna under driftsättningen. Skapa inkommande SSH-regler:

   - Privat: skapa inkommande regel för att tillåta SSH från Offentlig
   - Offentlig: skapa inkommande regel för att tillåta SSH från Privat och från Offentlig

2. Använd pem-nyckeln som du skapade i steg 1.B för att ansluta till bastionvärden:

På Mac-terminalen:

Kör följande kommandon från katalogen där pem-nyckeln finns lagrad:

```
ssh-add –apple-use-keychain <keyName>.pem
```

If you get a warning about private key being accessible by others, then run this command: `chmod 600 <keyName.pem>` and then run the `ssh-add` command again.

Connect to the bastion host with this command: `ssh -A ec2-user@IPaddress`

For example: `ssh -A ec2-user@3.15.12.112.`

På Windows med PuTTY och Pageant:

a. Skapa en ppk-fil från pem-nyckeln: använd PuTTY Key Generator. Ladda pem-nyckeln som skapades i steg 1.B. Klicka på **Spara privat nyckel** efter import av nyckeln. Detta skapar en ppk-fil.

b. I PuTTY – öppna konfigurationen och gör följande ändringar:

   - Sessioner > Värdnamn: lägg till IP-adressen för bastionvärden.
   - Sessioner > Port: 22
   - Anslutning > Data > Användarnamn för automatisk inloggning: ec2-user
   - Anslutning > SSH > Auth > Tillåt agentvidarebefordran
   - Anslutning > SSH > Auth > För privat nyckel – klicka på Bläddra och välj .ppk-filen som skapades.

c. Installera Pageant och ladda ppk-filen i applikationen.

Steg 5 - Installera PostgreSQL

Terraform-mallen installerar inte PostgreSQL för att användas som den externa lagringsplatsen. Den associerade säkerhetsgruppen och subnätet skapas dock. Om den externa lagringsplatsen ska installeras på en EC2-instans som kör PostgreSQL

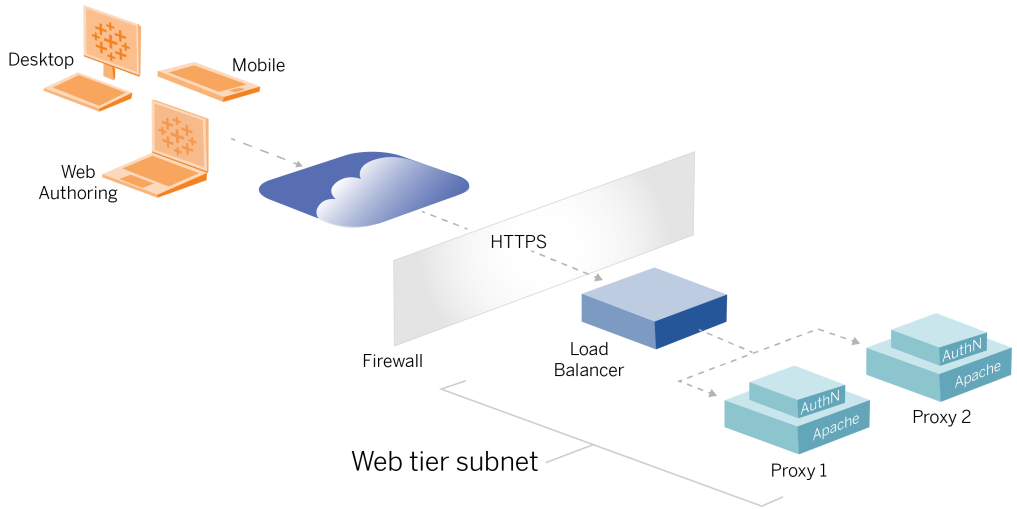
måste EC2-instansen driftsättas enligt beskrivningen i Del 3 – förbereda för företagsdriftsättning av Tableau Server.

Installera och konfigurera PostgreSQL samt säkerhetskopiera .tar enligt beskrivningen i Del 4 – Installera och konfigurera Tableau Server.

Steg 6 – (valfritt) kör DeployTab4EDG

Skriptet TabDeploy4EDG automatiserar implementeringen av Tableau-driftsättningen med fyra noder som beskrivs i del 4. Se TabDeploy4EDG – automatiserat installationsskript.

Bilaga - Exempel på driftsättning på webbnivå med Apache



Detta ämne ger dig en procedur från början till slut som beskriver hur du implementerar webbnivån i exemplet med AWS-referensarkitektur. En exempelkonfiguration består av följande komponenter:

- Lastbalanserare för AWS-programmet
- Apache-proxyservrar
- Mellon-autentiseringsmodul
- IdP för Okta
- SAML-autentisering

Obs! Det exempel på webbnivåkonfiguration som visas i det här avsnittet innehåller detaljerade procedurer för att distribuera programvara och tjänster från tredje part. Vi har gjort vårt bästa för att verifiera och dokumentera procedurerna för att möjliggöra webbnivåscenariot. Programvara från tredje part kan dock ändras eller så kan ditt

scenario skilja sig från den referensarkitektur som beskrivs här. Se dokumentationen från tredje part för konfigurationsinformation och support som har företräde.

Linux-exemplen i detta avsnitt visar kommandon för RHEL-liknande distributioner. Mer specifikt har kommandona här utvecklats med Amazon Linux 2-distributionen. Om du kör Ubuntu-driftsättningen redigerar du kommandona på lämpligt sätt.

Driftsättning av webbnivån i det här exemplet följer en stegvis konfigurations- och verifieringsprocedur. Konfigurationen av kärnwebbnivån består av följande steg för att aktivera HTTP mellan Tableau och internet. Apache körs och konfigureras för omvänd proxy/lastbalansering bakom lastbalanserare för AWS-programmet:

1. Installera Apache
2. Konfigurera omvänd proxy för att testa anslutning till Tableau Server
3. Konfigurera lastbalansering på proxyn
4. Konfigurera lastbalanserare för AWS-program

Efter att webbnivån konfigurerats och anslutning med Tableau verifierats så konfigurerar du autentisering med den externa leverantören.

Installera Apache

Kör följande procedur på bägge EC2-värdar (Proxy 1 och Proxy 2). Om du distribuerar i AWS enligt referensarkitekturexemplet så borde du ha två tillgänglighetszoner där en enskild proxyserver körs i varje zon.

1. Installera Apache:

   ```
   sudo yum update -y
   sudo yum install -y httpd
   ```

2. Konfigurera för att starta Apache vid omstart:

```
sudo systemctl enable --now httpd
```

3. Verifiera att den version av httpd du har installerad inkluderar `proxy_hcheck_module`:

   ```
   sudo httpd -M
   ```

   `proxy_hcheck_module` krävs. Om din version av httpd inte inkluderar den här modulen så uppdaterar du till en version av httpd som inkluderar den.

Konfigurera proxy för att testa anslutning till Tableau Server

Kör den här proceduren på en av proxyvärdarna (Proxy 1). Syftet med det här steget är att verifiera anslutningen mellan internet till din proxyserver till Tableau Server i den privata säkerhetsgruppen.

1. Skapa en fil med namnet `tableau.conf` och lägg till den i katalogen `/etc/httpd/conf.d`.

   Kopiera följande kod och ange nycklarna `ProxyPass` och `ProxyPassReverse` med den privata IP-adressen för Tableau Server nod 1.

   > **Viktigt:** Den konfiguration som visas nedan är inte säker och bör inte användas i produktion. Konfigurationen bör endast användas under installationsprocessen för att verifiera anslutning från slutpunkt till slutpunkt.

   Om den privata IP-adressen för nod 1 till exempel är `10.0.30.32` så skulle innehållet i filen `tableau.conf` vara:

   ```
   <VirtualHost *:80>
   ProxyPreserveHost On
   ```

```
ProxyPass "/" "http://10.0.30.32:80/"
ProxyPassReverse "/" "http://10.0.30.32:80/"
</VirtualHost>
```

2. Starta om httpd:

   ```
   sudo systemctl restart httpd
   ```

Verifiering: bastopologikonfiguration

Du borde kunna komma åt adminsidan för Tableau Server genom att gå till `http://<proxy-public-IP-address>`.

Om inloggningssidan för Tableau Server inte öppnas i din webbläsare så kan du följa följande steg på Proxy 1-värden:

- Stoppa och starta om httpd som ett första felsökningssteg.
- Dubbelkolla filen `tableau.conf`. Verifiera att den privata IP-adressen till nod 1 stämmer. Verifiera dubbla citattecken och kontrollera syntax noggrant.
- Kör `curl`-kommandot på den omvända proxyservern med den privata IP-adressen för nod 1, till exempel `curl 10.0.1.90`. Om kommandotolken inte returnerar html eller om den returnerar html för Apache-testwebbsidan så kan du verifiera protokoll-/portkonfigurationen mellan de offentliga och privata säkerhetsgrupperna.
- Kör `curl`-kommandot med den privata IP-adressen för proxy 1, till exempel `curl 10.0.0.163`. Om kommandotolken returnerar html-kod för Apache-testwebbsidan så har proxyfilen inte konfigurerats korrekt.
- Starta alltid om httpd (`sudo systemctl restart httpd`) efter konfigurationsändringar av proxyfilen eller säkerhetsgrupper.
- Se till att TSM körs på nod 1.

Konfigurera lastbalansering på proxyn

1. På samma proxyvärd (Proxy 1) där du skapade `tableau.conf`-filen tar du bort den befintliga konfigurationen för virtuell värd och redigerar filen så att den inkluderar logik för lastbalansering.

   Exempel:

```
<VirtualHost *:80>
ServerAdmin admin@example.com
#Load balancing logic.
ProxyHCExpr ok234 {%{REQUEST_STATUS} =~ /^[234]/}
Header add Set-Cookie "ROUTEID=.%{BALANCER_WORKER_ROUTE}e;
path=/" env=BALANCER_ROUTE_CHANGED
<Proxy balancer://tableau>
#Replace IP addresses below with the IP addresses to the
Tableau Servers running the Gateway service.
BalancerMember http://10.0.3.40/ route=1 hcmethod=GET
hcexpr=ok234 hcuri=/favicon.ico
BalancerMember http://10.0.4.151/ route=2 hcmethod=GET
hcexpr=ok234 hcuri=/favicon.ico
ProxySet stickysession=ROUTEID
</Proxy>
ProxyPreserveHost On
ProxyPass / balancer://tableau/
ProxyPassReverse / balancer://tableau/
</VirtualHost>
```

2. Stoppa och starta om httpd:

```
sudo systemctl stop httpd
sudo systemctl start httpd
```

3. Kontrollera konfigurationen genom att gå till den offentliga IP-adressen för proxy 1.

Kopiera konfigurationen till den andra proxyservern

1. Kopiera `tableau.conf`-filen från proxy 1 och spara den till katalogen `/etc/httpd/conf.d` på proxy 2-värden.

2. Stoppa och starta om httpd:

```
sudo systemctl stop httpd
sudo systemctl start httpd
```

3. Kontrollera konfigurationen genom att gå till den offentliga IP-adressen för proxy 2.

Konfigurera lastbalanserare för AWS-program

Konfigurera belastningsutjämnaren som en HTTP-lyssnare. Här beskrivs hur du lägger till en belastningsutjämnare i AWS:

Steg 1: Skapa målgrupp

Målgruppen är en AWS-konfiguration där de EC2-instanser som körs för dina proxyservrar definieras. Dessa är målen för LBS-trafik.

1. EC2 >**Målgrupper** > **Skapa målgrupp**
2. På sidan Skapa:
   - Ange ett namn på målgruppen, till exempel `TG-internal-HTTP`
   - Måltyp: instanser
   - Protokoll: HTTP
   - Port: 80
   - VPC: Välj VPC
   - Under **Hälsokontroller** > **Avancerade inställningar för hälsokontroller** > **Framgångskoder** lägger du till kodlistan enligt följande: `200,303`.
   - Klicka på **Skapa**
3. Välj den målgrupp som du just skapade och klicka sedan på fliken **Mål**:
   - Klicka på **Redigera**.
   - Välj de EC2-instanser (eller en instans om du konfigurerar en i taget) som kör proxyprogram och klicka sedan på **Lägg till bland registrerade**.
   - Klicka på **Spara**.

Steg 2: Starta guiden för belastningsutjämnaren

1. EC2 > **Belastningsutjämnare** > **Create Load Balancer** (Skapa belastningsutjämnare)
2. Skapa en belastningsutjämnare för program på sidan ”Välj typ av belastningsutjämnare”.

Obs! Gränssnittet som visas för konfigurering av belastningsutjämnaren är inte samma på alla AWS-datacenter. Via stegen i Guidekonfiguration nedan får du åtkomst till AWS-konfigurationsguiden som börjar med **steg 1 Konfigurera belastningsutjämnare**.

Om alla konfigurationer visas på samma sida i datacentret där knappen **Skapa belastningsutjämnare** visas längst ned följer du proceduren ”Konfiguration för enskild sida” nedan.

Guidekonfiguration

1. Sidan **Konfigurera belastningsutjämnare**:
   - Ange namn
   - Schema: internetanpassat (standard)
   - IP-adresstyp: ipv4 (standard)
   - Lyssnare (lyssnare och dirigering):
     a. Lämna HTTP-standardlyssnaren
     b. Klicka på **Lägg till lyssnare** och lägg till `HTTPS:443`
   - VPC: välj den VPC där du har installerat allt
   - Tillgänglighetszoner:
     - Välj **a** och **b** för dina datacenterregioner
     - I varje motsvarande listruta väljer du det offentliga undernätet (där dina proxyservrar finns).
   - Klicka på **Configure Security Settings** (Konfigurera säkerhetsinställningar)
2. Sidan **Konfigurera säkerhetsinställningar**

- Ladda upp ditt offentliga SSL-certifikat.
- Klicka på **Next: Configure Security Groups** (Nästa: Konfigurera säkerhetsgrupper).

3. Sidan **Konfigurera säkerhetsgrupper**:
   - Välj den offentliga säkerhetsgruppen. Om standardvalet för säkerhetsgrupp väljs, ska valet rensas.
   - Klicka på **Next: Configure Routing** (Nästa: Konfigurera dirigering).
4. Sidan **Configure Routing** (Konfigurera dirigering).
   - Målgrupp: Befintlig målgrupp.
   - Namn: Välj den målgrupp som du skapade tidigare.
   - Klicka på **Next: Register Targets** (Nästa: Registrera mål).
5. Sidan **Register Targets** (Registrera mål)
   - Du borde se de två proxyserverinstanserna som du konfigurerade tidigare.
   - Klicka på **Next: Review** (Nästa: Granska).
6. Sidan **Review** (Granska)

   Klicka på **Skapa**.

Konfiguration för enskild sida

Grundläggande konfiguration

- Ange namn
- Schema: internetanpassat (standard)
- IP-adresstyp: ipv4 (standard)

Nätverkskartläggning

- VPC: välj den VPC där du har installerat allt
- Kartläggningar:
  - Välj tillgänglighetszoner **a** och **b** (eller liknande) för dina datacenterregioner
  - I varje motsvarande listruta väljer du det offentliga undernätet (där dina proxyservrar finns).

Säkerhetsgrupper

Välj den offentliga säkerhetsgruppen. Om standardvalet för säkerhetsgrupp väljs, ska valet rensas.

Lyssnare och dirigering

- Lämna HTTP-standardlyssnaren. För **Standardåtgärd** anger du Målgruppen som du ställde in tidigare.
- Klicka på **Lägg till lyssnare** och lägg till `HTTPS:443`. För **Standardåtgärd** anger du Målgruppen som du ställde in tidigare.

Skydda lyssnarinställningarna

- Ladda upp ditt offentliga SSL-certifikat.

Klicka på **Create load balancer** (Skapa belastningsutjämnare).

Steg 3: Aktivera varaktighet

1. När belastningsutjämnaren har skapats ska varaktigheten aktiveras i målgruppen.
   - Öppna sidan AWS-målgrupp (**EC2** > **Belastningsutjämning** > **Målgrupper**) och välj den målgruppsinstans som du just konfigurerade. I **Åtgärdsmenyn** väljer du **Redigera attribut**.
   - På sidan **Redigera attribut** väljer du **Varaktighet**, anger en varaktigheten `1 day` och trycker sedan på **Spara ändringar**.
2. Aktivera varaktighet för belastningsutjämnaren för HTTP-lyssnaren. Välj den belastningsutjämnare som du just konfigurerade och klicka sedan på fliken **Lyssnare**:
   - För **HTTP:80** klickar du på **Visa/redigera regler**. Öppna sidan **Regler** och klicka därefter på redigeringsikonen (en gång högst upp på sidan och sedan igen bredvid regeln) för att redigera regeln. Ta bort befintlig THEN-regel och ersätt

genom att klicka på **Lägg till åtgärd** > **Vidarebefordra till ...**. Ange den målgrupp som du har skapat i återstående DÅ-konfiguration. Aktivera varaktighet under Varaktighet på gruppnivå och ange varaktigheten till 1 dag. Spara inställningen och klicka sedan på **Uppdatera**.

Steg 4: Ställ in tidsgräns för inaktivitet för belastningsutjämnaren

Uppdatera tidsgränsen för inaktivitet till 400 sekunder för belastningsutjämnaren.

Välj den belastningsutjämnare du har konfigurerat för den här driftsättningen och klicka sedan på **Åtgärder** > **Redigera attribut**. Ställ in **tidsgränsen för inaktivitet** på `400` sekunder och klicka sedan på **Spara**.

Steg 5: Verifiera LBS-anslutning

Öppna sidan AWS-belastningsutjämnare (**EC2**> **Belastningsutjämnare**) och välj den belastningsutjämnare som du just konfigurerade.

Kopiera DNS-namnet som visas i **Beskrivning** och klistra in i webbläsaren för att komma åt inloggningssidan för Tableau Server.

Om ett 500-nivåfel uppstår måste du starta om proxyservrarna.

Uppdatera DNS med den offentliga Tableau-URL:en

Använd domänens DNS-zonnamn från beskrivningen för AWS-belastningsutjämnaren för att skapa ett CNAME-värde i din DNS. Trafik till din URL (tableau.example.com) borde skickas till ditt offentliga AWS DNS-namn.

Verifiera anslutning

Efter att dina DNS-uppdateringar slutförts borde du kunna gå till inloggningssidan för Tableau Server genom att ange din offentliga URL, till exempel `https://tableau.example.com`.

Exempel på autentiseringskonfiguration: SAML med extern IdP

Följande exempel beskriver hur du installerar och konfigurerar SAML med IdP för Okta och Mellon-autentiseringsmodulen för en Tableau-driftsättning som kör AWS-referensarkitekturen. Det här exemplet beskriver hur du konfigurerar Tableau Server och Apache-proxyservrar att använda HTTP. Okta skickar begäranden till AWS-belastningsutjämnaren över HTTPS men all intern trafik går över HTTP. När du konfigurerar för det här scenariot bör du vara medveten om HTTP- kontra HTTPS-protokollen när du anger URL-strängar.

Det här exemplet använder sig av Mellon som tjänstleverantörsmodul för förautentisering på de omvända proxyservrarna. Med den här konfigurationen ser du till att endast autentiserad trafik ansluter till Tableau Server, som även agerar som tjänsteleverantör med IdP för Okta. Därmed behöver du konfigurera två IdP-applikationer: en för Mellon-tjänsteleverantören och en för Tableau-tjänsteleverantören.

Skapa ett Tableau-administratörskonto

Ett vanligt misstag när man konfigurerar SAML är att inte skapa ett administratörskonto på Tableau Server innan SSO aktiveras.

Första steget är att skapa ett konto på Tableau Server med rollen som serveradministratör. För exemplet med Okta-scenariot måste användarnamnet vara i ett giltigt e-postadressformat, såsom användare@exempel.com. Du måste ange ett lösenord för den här användaren. Lösenordet kommer dock inte att användas efter att SAML har konfigurerats.

Konfigurera Okta-program med förautentisering

Scenariot från slutpunkt till slutpunkt som beskrivs i det här avsnittet kräver två Okta-program:

- Okta-program med förautentisering
- Okta Tableau Server-program

Var och en av dessa program är associerade med olika metadata som behöver konfigureras på den omvända proxyn och Tableau-servern, respektive.

Den här proceduren beskriver hur man skapar och konfigurerar Okta-program med förautentisering. Längre fram i detta ämne kommer du att skapa Okta Tableau Server-programmet. Se Oktas webbplats för utvecklare för ett kostnadsfritt Okta-testkonto med begränsade användare.

Skapa en SAML-appintegrering för Mellon-tjänsteleverantör med förautentisering.

1. Öppna Oktas administrationsöversikt > **Program** > **Skapa appintegrering**.
2. På sidan **Skapa en ny appintegrering** kan du välja **SAML 2.0** och sedan klicka på **Nästa**.
3. På fliken **Allmänna inställningar** ska du ange ett appnamn, såsom `Tableau Pre-Auth`, och sedan klicka på **Nästa**.
4. På fliken **Konfigurera SAML**:
    - URL för enkel inloggning. Det sista elementet av sökvägen i URL:en för enkel inloggning kallas för `MellonEndpointPath` i konfigurationsfilen `mellon.conf` som följer senare i den här proceduren. Du kan ange valfri slutpunkt. I det här exemplet är slutpunkten `sso`. Det sista elementet `postResponse` krävs : `https://tableau.example.com/sso/postResponse`.
    - Klicka ur kryssrutan: **Använd detta för URL för mottagaren och destinationen**.
    - URL för mottagaren: Samma som URL för SSO, men med HTTP. Till exempel `http://tableau.example.com/sso/postResponse`.
    - Mål-URL: samma som URL för enkel inloggning, men med HTTP. Till exempel `http://tableau.example.com/sso/postResponse`.
    - Audience URI (SP Entity ID). Till exempel `https://tableau.example.com`.
    - Format på namn-ID: `EmailAddress`
    - Användarnamn i programmet: `Email`

- Attribututlåtanden: namn = `mail`; namnformat = `Unspecified`; värde = `user.email`.

Klicka på **Nästa**.

5. Välj följande på fliken **Feedback**:
   - **Jag är Okta-kund och lägger till ett internt program**
   - **Det här är ett internt program som vi har skapat**
   - Klicka på **Slutför**.
6. Skapa metadatafilen för tjänsteleverantörens förautentisering:
   - I Okta: **Program** > **Program** > Ditt nya program (t.ex. `Tableau Pre-Auth`) > **Logga in**
   - Bredvid **SAML-signeringscertifikat** klickar du på **Visa anvisningar om SAML-konfiguration**.
   - På sidan **Så här konfigurerar du SAML 2.0 för <pre-auth>-program** rullar du nedåt till delen **Valfritt** , **Uppge följande IdP-metadata för serviceleverantören**.
   - Kopiera innehållet i XML-fältet och spara det i en fil med namnet `pre-auth_idp_metadata.xml`.
7. (Valfritt) Konfigurera multifaktorautentisering:
   - I Okta: **Program** > **Program** > Ditt nya program (t.ex. `Tableau Pre-Auth`) > **Logga in**
   - Klicka på **Lägg till regel** under **Inloggningspolicy**.
   - Ange ett namn och de olika MFA-alternativen i **Programmets inloggningsregel**. Du kan lämna alla alternativ som standard för att testa funktionaliteten. Under **Åtgärder** måste du dock välja **Fråga efter faktor** och sedan ange hur ofta användare måste logga in. Klicka på **Spara**.

Skapa och tilldela Okta-användare

1. Skapa en användare, i Okta, med samma användarnamn som skapades i Tableau (användare@example.com): **Katalog** > **Personer** > **Lägg till person**.

2. Tilldela dett nya Okta-programmet till den personen när användaren har skapats: Klicka på användarnamnet och tilldela sedan programmet i **Tilldela program**.

Installera Mellon för förautentisering

1. På de EC2-instanser som kör Apache-proxyservern kör du följande kommandon för att installera PHP- och Mellon-moduler:

   ```
   sudo yum install httpd php mod_auth_mellon
   ```

2. Skapa katalogen `/etc/httpd/mellon`

Konfigurerar Mellon som förautentiseringsmodul

Kör den här proceduren på båda proxyservrarna.

Du måste ha en kopia av `pre-auth_idp_metadata.xml`-filen som du skapade från Okta-konfigurationen.

1. Ändra katalog:

   ```
   cd /etc/httpd/mellon
   ```

2. Skapa metadata för tjänsteleverantören. Kör skriptet `mellon_create_metadata.sh`. Du måste inkludera entitets-ID och retur-URL för din organisation i kommandot.

   Retur-URL kallas för *URL för enkel inloggning* i Okta. Det slutliga elementet för sökvägen i retur-URL:en kallas för `MellonEndpointPath` i konfigurationsfilen `mellon.conf` som kommer senare i den här proceduren. I det här exemplet anger vi `sso` som slutpunktssökväg.

   Exempel:

```
sudo /usr/libexec/mod_auth_mellon/mellon_create_metadata.sh
https://tableau.example.com "https://tableau.example.com/sso"
```

Skriptet returnerar tjänsteleverantörens certifikat, nyckel och metadatafiler.

3. Byt namn på tjänsteleverantörens filer i katalogen `mellon` för ökad läsbarhet. Vi kallar dessa filer följande namn i dokumentationen:

```
sudo mv *.key mellon.key
sudo mv *.cert mellon.cert
sudo mv *.xml sp_metadata.xml
```

4. Kopiera `pre-auth_idp_metadata.xml`-filen till samma katalog.

5. Skapa filen `mellon.conf` i katalogen `/etc/httpd/conf.d`:

```
sudo nano /etc/httpd/conf.d/mellon.conf
```

6. Kopiera följande innehåll till `mellon.conf`.

```
<Location />
MellonSPPrivateKeyFile /etc/httpd/mellon/mellon.key
MellonSPCertFile /etc/httpd/mellon/mellon.cert
MellonSPMetadataFile /etc/httpd/mellon/sp_metadata.xml
MellonIdPMetadataFile /etc/httpd/mellon/pre-auth_idp_
metadata.xml
MellonEndpointPath /sso
MellonEnable "info"
</Location>
```

7. Lägg till följande innehåll till den befintliga `tableau.conf`-filen:

Inuti blocket `<VirtualHost *:80>` lägger du till följande innehåll. Uppdatera `ServerName` med det offentliga värdnamnet i din entitets-ID:

```
DocumentRoot /var/www/html
ServerName tableau.example.com
```

```
ServerSignature Off
ErrorLog logs/error_sp.log
CustomLog logs/access_sp.log combined
LogLevel info
```

Lägg till platsblocket utanför blocket `<VirtualHost *:80>`. Uppdatera `MellonCookieDomain` med toppnivådomänen för att bevara cookie-informationen så som visas:

```
<Location />
AuthType Mellon
MellonEnable auth
Require valid-user
MellonCookieDomain example.com
</Location>
```

Den färdiga `tableau.conf`-filen borde se ut som i följande exempel:

```
<VirtualHost *:80>
ServerAdmin admin@example.com
ProxyHCExpr ok234 {%{REQUEST_STATUS} =~ /^[234]/}
Header add Set-Cookie "ROUTEID=.%{BALANCER_WORKER_ROUTE}e;
path=/" env=BALANCER_ROUTE_CHANGED
<Proxy balancer://tableau>
BalancerMember http://10.0.3.36/ route=1 hcmethod=GET
hcexpr=ok234 hcuri=/favicon.ico
BalancerMember http://10.0.4.15/ route=2 hcmethod=GET
hcexpr=ok234 hcuri=/favicon.ico
ProxySet stickysession=ROUTEID
</Proxy>
ProxyPreserveHost On
ProxyPass / balancer://tableau/
ProxyPassReverse / balancer://tableau/
DocumentRoot /var/www/html
ServerName tableau.example.com
ServerSignature Off
```

```
ErrorLog logs/error_sp.log
CustomLog logs/access_sp.log combined
LogLevel info
</VirtualHost>
<Location />
AuthType Mellon
MellonEnable auth
Require valid-user
MellonCookieDomain example.com
</Location>
```

8. Verifiera konfigurationen. Kör följande kommando:

   ```
   sudo apachectl configtest
   ```

   Om konfigurationstestet returnerar ett fel, åtgärdar du felen och kör konfigurationstestet igen. En lyckad konfiguration returnerar `Syntax OK`.

9. Starta om httpd:

   ```
   sudo systemctl restart httpd
   ```

Skapa Tableau Server-applikation i Okta

1. På instrumentpanelen för Okta: **Program** > **Program** > **Browse App Catalog** (Bläddra i appkatalog).
2. Sök efter `Tableau` i **Browse App Integration Catalog** (Bläddra i appintegreringskatalog), välj panelen för Tableau Server och klicka på **Lägg till**.
3. På **Add Tableau Server** (Lägg Tableau Server) > **General Settings** (Allmänna inställningar) anger du en etikett och klickar sedan på **Nästa**.
4. Välj **SAML 2.0** i Sign-On Options (Inloggningsalternativ) och rulla ned till Advanced Sign-on Settings (Avancerade inloggningsinställningar):
   - **SAML Entity ID** (Entitets-ID för SAML): Ange den offentliga URL:en, t.ex. https://tableau.example.com.
   - **Application user name format** (Format för användarnamn för programmet): E-postadress

5. Starta en webbläsare genom att klicka på länken **Identity Provider metadata** (Metadata för identitetsprovider). Kopiera webbläsarlänken. Det här är länken du använder när du konfigurerar Tableau i stegen som följer.
6. Klicka på **Klart**.
7. Tilldela användaren (användare@företag.com) den nya Tableau Server Okta-appen: Klicka på användarnamnet och tilldela sedan programmet i **Assign Application** (Tilldela program).

Aktivera SAML på Tableau Server för IdP

Kör den här proceduren på Tableau Server-nod 1.

1. Hämta metadata för Tableau Server-programvaran från Okta. Använd länken som du sparade från föregående procedur.

   ```
   wget https://dev-66144217.okta.com/app/exk1egxgt1fhjkSeS5d7/sso/saml/metadata -O idp_metadata.xml
   ```

2. Kopiera ett TLS-certifikat och den relaterade nyckelfilen till Tableau Server. Nyckelfilen måste vara en RSA-nyckel. Mer information om SAML-certifikat och IdP-krav finns i *SAML-krav* (Linux).

   För att underlätta certifikathanteringen och driftsättningen, och som en rekommenderad säkerhetsåtgärd, bör du använda certifikat genererade av någon av de stora betrodda certifikatutfärdarna (CA). Du kan också generera självsignerade certifikat eller använda certifikat från en PKI för TLS.

   Om du inte har något TLS-certifikat kan du skapa ett självsignerat certifikat med hjälp av den inbäddade proceduren nedan.

Skapa ett självsignerat certifikat

Kör den här proceduren på Tableau Server-nod 1.

a. Generera signeringsnyckel för rotcertifikatutfärdare (CA, Certificate Authority):

```
openssl genrsa -out rootCAKey-saml.pem 2048
```

b. Skapa certifikat för rot-CA:

```
openssl req -x509 -sha256 -new -nodes -key rootCAKey-
saml.pem -days 3650 -out rootCACert-saml.pem
```

Du uppmanas att ange värden i certifikatfälten. Exempel:

```
Country Name (2 letter code) [XX]:US
State or Province Name (full name) []:Washington
Locality Name (eg, city) [Default City]:Seattle
Organization Name (eg, company) [Default Company
Ltd]:Tableau
Organizational Unit Name (eg, section) []:Operations
Common Name (eg, your name or your server's hostname)
[]:tableau.example.com
Email Address []:example@tableau.com
```

c. Skapa certifikatet och relaterad nyckel (`server-saml.csr` och `server-saml.key` i nedanstående exempel). Certifikatmottagarens namn måste stämma överens med det offentliga värdnamnet för Tableau-värden. Certifikatmottagarens namn anges med alternativet `-subj` i formatet "`/CN=<host-name>`", till exempel:

```
openssl req -new -nodes -text -out server-saml.csr -keyout
server-saml.key -subj "/CN=tableau.example.com"
```

d. Signera det nya certifikatet med CA-certifikatet som du skapade ovan. Följande kommando skapar även certifikatet i `crt`-format:

```
openssl x509 -req -in server-saml.csr -days 3650 -CA
rootCACert-saml.pem -CAkey rootCAKey-saml.pem -
```

```
CAcreateserial -out server-saml.crt
```

e. Konvertera nyckelfilen till RSA. Tableau behöver en RSA-nyckelfil för SAML. Kör följande kommando för att konvertera nyckeln:

```
openssl rsa -in server-saml.key -out server-saml-rsa.key
```

3. Konfigurera SAML. Kör följande kommando och ange ditt enhets-ID och din retur-URL, samt sökvägarna till metadatafilen, certifikatfilen och nyckelfilen:

```
tsm authentication saml configure --idp-entity-id
"https://tableau.example.com" --idp-return-url
"https://tableau.example.com" --idp-metadata idp_metadata.xml -
-cert-file "server-saml.crt" --key-file "server-saml-rsa.key"

tsm authentication saml enable
```

4. Om din organisation använder Tableau Desktop 2021.4 eller senare måste du köra följande kommando för att aktivera autentisering via omvända proxyservrar.

   Tableau Desktop-versioner 2021.2.1 – 2021.3 kommer att fungera utan att köra detta kommando, förutsatt att din förautentiseringsmodul (t.ex. Mellon) har konfigurerats för att tillåta att domäncookies bevaras på toppnivå.

```
tsm configuration set -k features.ExternalBrowserOAuth -v false
```

5. Tillämpa konfigurationsändringar:

```
tsm pending-changes apply
```

Validera SAML-funktion

Validera SAML-funktion från slutpunkt till slutpunkt genom att logga in på Tableau Server med den offentliga URL:en (t.ex. https://tableau.example.com) med det Tableau-administratörskonto som du skapade i början av den här proceduren.

Felsökning av validering

Dålig begäran: Ett vanligt fel för det här scenariot är ett Dålig begäran-fel från Okta. Det här problemet uppstår ofta när webbläsaren cachelagrar data från tidigare Okta-sessioner. Om du till exempel hanterar Okta-applikationer som en Okta-administratör och därefter försöker få åtkomst till Tableau med ett annat Okta-aktiverat konto så kan sessionsdata från administratörsdata orsaka Dålig begäran-felet. Om felet kvarstår även efter att du rensat den lokala webbläsarens cacheminne så kan du testa att validera Tableau-scenariot genom att ansluta med en annan webbläsare.

En annan orsak till felet ”Felaktig begäran” är ett stavfel i en av de många URL:er som du anger under Okta-, Mellon- och SAML-konfigurationsprocesserna. Kontrollera alla dessa noggrant.

Ofta anger httpd-filen `error.log` på Apache-servern vilken URL som orsakar felet.

Hittades inte – Den begärda URL:en hittades inte på den här servern: Det här felet indikerar ett av flera konfigurationsfel.

Om användaren autentiserats med Okta och därefter stöter på det här felet så har du sannolikt laddat upp Okta förautentiseringsapplikationen till Tableau Server när du konfigurerade SAML. Verifiera att du har konfigurerat Okta-applikationsmetadata för Tableau Server på Tableau Server och inte applikationsmetadata för Okta förautentisering.

Andra felsökningssteg:

- Granska `tableau.conf` och se om det finns stavfel eller konfigurationsfel
- Granska applikationsinställningarna för Okta förautentisering. Se till att HTTP- kontra HTTPS-protokollen angetts som specificerade i det här ämnet.
- Starta om httpd på bägge proxyservrar.
- Verifiera att `sudo apachectl configtest` returnerar Syntax OK för bägge proxyservrar.
- Verifiera att testanvändaren tilldelats båda applikationerna i Okta.
- Verifiera att ihållande angetts för lastbalanseraren och associerade målgrupper.

Konfigurera SSL/TLS från belastningsutjämnare till Tableau Server

Vissa organisationer kräver en komplett (”end-to-end”) krypteringskanal från klient till backend-tjänst. Standardreferensarkitekturen som beskrivs under denna punkt specificerar SSL från klienten till den belastningsutjämnare som körs på webbnivån i din organisation.

För att konfigurera SSL från lastbalanserare till Tableau Server måste du göra följande:

- Installera ett giltigt SSL-certifikat på både Tableau- och proxyservrarna.
- Konfigurera SSL från lastbalanseraren till omvända proxyservrar.
- Konfigurera SSL från proxyservrarna till Tableau Server.
- Du kan också konfigurera SSL från Tableau Server till PostgreSQL-instansen.

Resten av detta ämne beskriver denna implementering inom ramarna för exempelreferensarkitekturen för AWS.

Exempel: Konfigurera SSL/TLS i AWS-referensarkitektur

Det här avsnittet beskriver hur du konfigurerar SSL på Tableau och konfigurerar SSL på en Apache-proxyserver, allt som körs i AWS-referensarkitektur.

Linux-procedurerna i detta exempel visar kommandon för RHEL-liknande distributioner. Mer specifikt har kommandona här utvecklats med Amazon Linux 2-distributionen. Om du kör Ubuntu-driftsättningen redigerar du kommandona på lämpligt sätt.

Steg 1: Ta reda på certifikat och relaterade nycklar

För att underlätta certifikathanteringen och driftsättningen, och som en rekommenderad säkerhetsåtgärd, bör du använda certifikat genererade av någon av de stora betrodda certifikatutfärdarna (CA).

Du kan också generera självsignerade certifikat eller använda certifikat från en PKI för TLS.

Följande procedur visar hur man genererar självsignerade certifikat. Om du använder certifikat från tredje part som vi rekommenderar kan du hoppa över den här proceduren.

Kör den här proceduren på en av proxyvärdarna. När du har genererat certifikatet och tillhörande nyckel kommer du att dela detta med den andra proxyvärden och Tableau Server-nod 1.

1. Generera signeringsnyckel för rotcertifikatutfärdare (CA, Certificate Authority):

   ```
   openssl genrsa -out rootCAKey.pem 2048
   ```

2. Skapa certifikat för rot-CA:

   ```
   openssl req -x509 -sha256 -new -nodes -key rootCAKey.pem -days
   3650 -out rootCACert.pem
   ```

   Du uppmanas att ange värden i certifikatfälten. Exempel:

   ```
   Country Name (2 letter code) [XX]:US
   State or Province Name (full name) []:Washington
   Locality Name (eg, city) [Default City]:Seattle
   Organization Name (eg, company) [Default Company Ltd]:Tableau
   Organizational Unit Name (eg, section) []:Operations
   Common Name (eg, your name or your server's hostname)
   []:tableau.example.com
   Email Address []:example@tableau.com
   ```

3. Skapa certifikatet och relaterad nyckel (`serverssl.csr` och `serverssl.key` i nedanstående exempel). Certifikatmottagarens namn måste stämma överens med det offentliga värdnamnet för Tableau-värden. Certifikatmottagarens namn anges med alternativet `-subj` i formatet `"/CN=<host-name>"`, till exempel:

   ```
   openssl req -new -nodes -text -out serverssl.csr -keyout
   serverssl.key -subj "/CN=tableau.example.com"
   ```

4. Signera det nya certifikatet med det CA-certifikat som du skapade i steg 2. Följande kommando skapar även certifikatet i `crt`-format:

   ```
   openssl x509 -req -in serverssl.csr -days 3650 -CA
   rootCACert.pem -CAkey rootCAKey.pem -CAcreateserial -out
   serverssl.crt
   ```

Steg 2: Konfigurera proxyserver för SSL

Kör den här proceduren på båda proxyservrarna.

1. Installera SSL-modulen för Apache:

   ```
   sudo yum install mod_ssl
   ```

2. Skapa katalogen `/etc/ssl/private`:

   ```
   sudo mkdir -p /etc/ssl/private
   ```

3. Kopiera crt- och nyckelfilerna till följande `/etc/ssl/`-sökvägar:

   ```
   sudo cp serverssl.crt /etc/ssl/certs/

   sudo cp serverssl.key /etc/ssl/private/
   ```

4. Uppdatera befintlig `tableau.conf` med följande uppdateringar:

   - Lägg till SSL-omskrivningsblocket:

     ```
     RewriteEngine on
     RewriteCond %{SERVER_NAME} =tableau.example.com
     RewriteRule ^ https://%{SERVER_NAME}%{REQUEST_URI}
     [END,NE,R=permanent]
     ```

   - I SSL-omskrivningsblocket uppdaterar du servernamnet `RewriteCond`: Lägg till ditt offentliga värdnamn, till exempel `tableau.example.com`
   - Ändra `<VirtualHost *:80>` till `<VirtualHost *:443>`.

- Omslut blocken `<VirtualHost *:443>` och `<Location />` med `<IfModule mod_ssl.c>` ... `</IfModule>`.
- `BalancerMember`: Ändra protokollet från `http` till `https`.
- Lägg till `SSL*`-element inuti `<VirtualHost *:443>`-blocket:

```
SSLEngine on
SSLCertificateFile /etc/ssl/certs/serverssl.crt
SSLCertificateKeyFile /etc/ssl/private/serverssl.key
SSLProxyEngine on
SSLProxyVerify none
SSLProxyCheckPeerName off
SSLProxyCheckPeerExpire off
```

- I `LogLevel`-elementet: Lägg till `ssl:warn`.
- Valfritt: Om du har installerat och konfigurerat en autentiseringsmodul kan du ha ytterligare element i filen tableau.conf. Till exempel kommer blocket `<Location /> </Location>` att innehålla element.

Här visas ett exempel på en tableau.conf-fil som konfigurerats för SSL:

```
RewriteEngine on
RewriteCond %{SERVER_NAME} =tableau.example.com
RewriteRule ^ https://%{SERVER_NAME}%{REQUEST_URI}
[END,NE,R=permanent]

<IfModule mod_ssl.c>
<VirtualHost *:443>
ServerAdmin admin@example.com
ProxyHCExpr ok234 {%{REQUEST_STATUS} =~ /^[234]/}
Header add Set-Cookie "ROUTEID=.%{BALANCER_WORKER_ROUTE}e;
path=/" env=BALANCER_ROUTE_CHANGED
<Proxy balancer://tableau>
BalancerMember https://10.0.3.36/ route=1 hcmethod=GET
hcexpr=ok234 hcuri=/favicon.ico
BalancerMember https://10.0.4.15/ route=2 hcmethod=GET
hcexpr=ok234 hcuri=/favicon.ico
ProxySet stickysession=ROUTEID
```

```
</Proxy>
ProxyPreserveHost On
ProxyPass / balancer://tableau/
ProxyPassReverse / balancer://tableau/
DocumentRoot /var/www/html
ServerName tableau.example.com
ServerSignature Off
ErrorLog logs/error_sp.log
CustomLog logs/access_sp.log combined
LogLevel info ssl:warn
SSLEngine on
SSLCertificateFile /etc/ssl/certs/serverssl.crt
SSLCertificateKeyFile /etc/ssl/private/serverssl.key
SSLProxyEngine on
SSLProxyVerify none
SSLProxyCheckPeerName off
SSLProxyCheckPeerExpire off
</VirtualHost>
<Location />
#If you have configured a pre-auth module (e.g. Mellon) include
those elements here.
</Location>
</IfModule>
```

5. Lägg till index.html fil för att undertrycka 403-fel:

   ```
   sudo touch /var/www/html/index.html
   ```

6. Starta om httpd:

   ```
   sudo systemctl restart httpd
   ```

Steg 3: Konfigurera Tableau Server för extern SSL

Kopiera filerna serverssl.crt och serverssl.key från Proxy 1-värddatorn till den ursprungliga Tableau Server (nod 1).

Kör följande kommandon på nod 1:

```
tsm security external-ssl enable --cert-file serverssl.crt --key-
file serverssl.key
tsm pending-changes apply
```

Steg 4: Valfri autentiseringskonfiguration

Om du har konfigurerat en extern identitetsleverantör för Tableau måste du sannolikt uppdatera returadresser (URL) i den administrativa instrumentpanelen för IdP.

Om du till exempel använder ett förauktoriseringsprogram för Okta måste du uppdatera programmet för att använda HTTPS-protokollet för mottagaradressen och destinationsadressen.

Steg 5: Konfigurera AWS-lastbalanserare för HTTPS

Om du driftsätter med AWS-lastbalanserare enligt denna guide konfigurerar du om AWS-lastbalanseraren så att denna skickar HTTPS-trafik till proxyservrarna:

1. Avregistrera befintlig HTTP-målgrupp:

   I **Målgrupper** väljer du den HTTP-målgrupp som har konfigurerats för lastbalanseraren, klickar på **Åtgärder** och sedan på **Registrera och avregistrera instans**.

   På sidan **Registrera och avregistrera mål** markerar du de instanser som för närvarande är konfigurerade, klickar på **Avregistrera** och sedan på **Spara**.

2. Skapa HTTPS-målgrupp:

   Målgrupper > **Skapa målgrupp**

   - Välj ”Instanser”
   - Ange ett namn på målgruppen, till exempel `TG-internal-HTTPS`
   - Välj din VPC
   - Protokoll: HTTPS 443
   - Under **Hälsokontroller** > **Avancerade inställningar för hälsokontroller** >

Framgångskoder lägger du till kodlistan enligt följande: `200,303`.
- Klicka på **Skapa**.

3. Välj den målgrupp som du just skapade och klicka sedan på fliken **Mål**:
    - Klicka på **Redigera**
    - Välj de EC2-instanser som kör proxyprogram och klicka sedan på **Lägg till bland registrerade**.
    - Klicka på **Spara**.

4. När målgruppen har skapats måste du aktivera varaktighet:
    - Öppna sidan AWS-målgrupp (**EC2** > **Belastningsutjämning** > **Målgrupper**) och välj den målgruppsinstans som du just konfigurerade. I **Åtgärdsmenyn** väljer du **Redigera attribut**.
    - På sidan **Redigera attribut** väljer du **Varaktighet**, anger en varaktigheten `1 day` och trycker sedan på **Spara ändringar**.

5. Uppdatera lyssnarreglerna vid belastningsutjämning. Välj den belastningsutjämnare du har konfigurerat för den här driftsättningen och klicka sedan på fliken **Lyssnare**.
    - För **HTTP:80** klickar du på **Visa/redigera regler**. Öppna sidan **Regler** och klicka därefter på redigeringsikonen (en gång högst upp på sidan och sedan igen bredvid regeln) för att redigera regeln. Ta bort den befintliga THEN-regeln och ersätt den genom att klicka på **Lägg till åtgärd** > **Omdirigera till ...**. I den resulterande THEN-konfigurationen anger du `HTTPS` och port `443` och låter övriga alternativ behålla standardinställningarna. Spara inställningen och klicka sedan på **Uppdatera**.
    - För **HTTP:443** klickar du på **Visa/redigera regler**. Öppna sidan **Regler** och klicka därefter på redigeringsikonen (en gång högst upp på sidan och sedan igen bredvid regeln) för att redigera regeln. I konfigurationen för **THen**, under **Vidarebefordra till ...** ändrar du målgruppen till den grupp som du nyss skapat. Aktivera varaktighet under **Varaktighet på gruppnivå** och ange varaktigheten till 1 dag. Spara inställningen och klicka sedan på **Uppdatera**.

Steg 6: Verifiera SSL

Verifiera konfigurationen genom att gå till https://tableau.example.com.