Tableau Server on Amazon Web Services

Deployment Guidelines and Best Practices

April 2017

Notices

This document is provided for informational purposes only. It represents AWS's product offerings and practices as of the publish date, which are subject to change without notice. Customers are responsible for making their own independent assessment of the information in this document and any use of AWS's products or services, each of which is provided "as is" without warranty of any kind, whether express or implied. This document does not create any warranties, representations, contractual commitments, conditions or assurances from AWS, its affiliates, suppliers or licensors. The responsibilities and liabilities of AWS to its customers are controlled by AWS agreements, and this document is not part of, nor does it modify, any agreement between AWS and its customers.

Table of Contents

Abstract

Amazon Web Services (AWS) provides a reliable, scalable, secure, and high performing infrastructure for the most demanding web applications—an infrastructure that matches IT costs with customer traffic patterns in real time.

As an enterprise business intelligence platform, Tableau Server provides comprehensive and robust capabilities for all aspects of business intelligence. These capabilities are quick and easy to deploy, and require minimal, if any, customizations. The core architecture of Tableau Server also runs two of the largest data platforms in the world: Tableau Online and Tableau Public.

This whitepaper provides technical guidance on how to deploy and configure Tableau Server on AWS. Additionally, it outlines key integrations and configurations unique to AWS products that enable you to use Tableau Server to best meet your needs.

Introduction

Tableau Server

Tableau Server provides business intelligence for organizations of all sizes and is built to simplify sharing and collaborating on interactive data visualizations by offering the following advantages:

- **Simple User Interface** – Tableau Server makes it easy to find, explore, and interact with analytic dashboards for every type of user. Powerful search capabilities and intuitive navigation controls make discovering content, users, and data sources straightforward.
- **Flexible data architecture** – If you have a fast database, Tableau Server can leverage that speed by maintaining live query connections back to that database. Alternatively, you can use Tableau Server to take in-memory snapshots of a data source (called extracts) and physically host that data on the Tableau Server platform.
- **Automatic data and content updates** – Tableau Server can refresh in-memory data extracts based on a set schedule, at specified intervals, or at incremental levels. You can also set alerts when data connections fail or use subscriptions to receive regularly scheduled emails about dashboards and reports.
- **Embedded analytics** – With Tableau Server, you can rapidly embed interactive dashboards within your organization's existing web portals. Built-in sharing capabilities quickly provide HTML snippets that you can use to place Tableau Server views directly into webpages, SharePoint portals, intranet wikis, and so on.

- **Scalable** – Tableau Server scales with both hardware and memory to support a growing organization. Flexible content management, user permissions, and detailed administrative capabilities make managing a growing Tableau Server platform a straightforward process.
- **Secure** – Tableau Server gives you security permissions at any level you need. With multi-tenancy, you can create multiple sites on the server to separate users and content. You can set individual permissions for projects, dashboards, or even users.
- **Mobile** – You can view a dashboard from anywhere, on any device. All dashboards are automatically optimized for mobile tablets with touch-sensitive UI without requiring any additional authoring or configuration.

Organizations need to keep up with the rapid changes required for global computing infrastructures. Additionally, they must find ways to deploy and deliver applications from distributed, cloud-based services with the confidence that these applications can deliver a consistent and reliable level of service, and can withstand significant, unpredictable spikes in traffic without missing a beat. Tableau Server, combined with the compute resources offered by AWS, meets these needs by providing analytics atop a global cloud infrastructure that drastically simplifies management.

Getting Started

Amazon Elastic Compute Cloud (Amazon EC2) provides resizable compute capacity in the cloud. Viewed from the framework of a traditional infrastructure, Amazon EC2 represents the servers that run your applications, web servers, databases, and, in this case, Tableau Server. You will deploy Tableau Server on one or more EC2 instances.

Basic Requirements and Recommendations

Tableau Server system requirements vary based on many factors, including the number of active users and the profile of your workload.

Tableau Server is available on the 64-bit Microsoft Windows platforms. Please visit our technical specifications page for the most up-to-date operating system requirements.

Note that Tableau Server minimum system requirements differ from what we recommend for best performance. Requirements and recommendations are also updated regularly. Therefore, we recommend you review the Online Help topic Minimum Hardware Requirements and Recommendations for Tableau Server for the most up-to-date information. Note that, to ensure you and your users to enjoy the best

performance possible on EC2, the guidance below suggests a

For a single server deployed to the EC2, we recommend an instance which provides sixteen vCPUs (the equivalent of eight CPU cores) and 64GiB of main memory.

> **Note:** We do not recommend running Tableau Server on an eight vCPU (four core) instance except for prototyping and testing.

Amazon EC2 allows you to provision a variety of instance types and sizes which provide different combinations of vCPU, memory, disk, and networking for hosting your Tableau server.

Typically, you should choose an instance type and size in accordance with the minimum recommendations (8 cores and 64GiB memory) for deploying Tableau Server on Amazon EC2. Currently, EC2 instances like the **m4.4xlarge** and **r4.4xlarge** meet our criteria for RAM and CPU. Either is a good starting point for deployment.

Deploying Tableau Server on EC2

Download Tableau Server from the Tableau download page (or through your customer portal if you are an existing customer) and install it on the Amazon EC2 instance of your choice.

You'll need to provide the right storage for use with Amazon EC2. **Amazon Elastic Block Store (Amazon EBS)** delivers persistent block-level storage volumes for use with EC2 instances in the AWS cloud.

We recommend deploying your EC2 instance with at least two volumes:

- A 30 – 50 GiB volume for the operating system
- A 100 GiB+ volume for Tableau Server

You should leverage **Amazon EBS General Purpose SSD** (GP2) volumes. Over the long term, we have generally experienced below-average-to-poor performance using magnetic disks and therefore recommend you avoid them.

A 100 GiB GP2 volume for Tableau Server supports a baseline performance of 300 IOPS with the ability to burst to 3,000 IOPS. This volume's ability to burst is important and requires a supply of I/O credits to do so.

If you exhaust I/O credits on a volume, disk latency will increase and performance will generally suffer as 300 IOPS typically will only support a light workload. We recommend that you monitor the following metrics in order to validate your storage is providing the requisite IOPS and throughput for Tableau:

- On Windows, the **Average Disk sec/Transfer** Performance Monitor counter should generally be below 18-20ms

- Use the AWS CloudWatch **Burst Balance** metric to watch for a close-to-exhausted I/O credit balance

For more information, see I/O Credits and Burst Performance in the Amazon EBS product documentation[1].

For heavy workloads where the storage subsystem must provide an absolute minimum number of IOPS for performance, you can create a larger **Amazon EBS General Purpose (SSD)** volume to get up to 10,000 IOPS with the ratio of 3 IOPS per GiB. A 1 TB volume will allow you to burst at 3000 IOPS indefinitely without ever exhausting your credit balance.

You may also leverage Amazon EBS Provisioned IOPS (SSD) volume with the IOPS you need. In either case, we recommend that you use Amazon EBS-optimized instances if you plan to run heavy extracts and processing.

Above and beyond choosing the correct type of instance and provisioning storage correctly, Tableau Server performance should be monitored and tuned as you would an on-premises server. Refer to the Improving Server Performance section in the Tableau Server documentation for comprehensive information to help you improve performance.

Automating Tableau Server Deployment with Tableau Quick Starts

Tableau Server Quick Starts utilize AWS CloudFormation templates to automatically deploy a standalone or multi-machine distributed (clustered) Tableau Server in your AWS account. The templates follow best practices from AWS and Tableau software for running Tableau on the cloud. For more information, see AWS Quick Starts: Tableau Server on AWS[3].

The appendix **Deploying Tableau Server on AWS** topic provides a step-by-step walk through of installing Tableau Server across multiple EC2 instance in a highly available and secure fashion. It closely mirrors the output of a multi-machine deployment created by the Quick Start.

Securing Tableau Server on AWS

AWS provides security features that Tableau Server can use to protect your environment, including the following:

- **Amazon Virtual Private Cloud (Amazon VPC)** adds another layer of network security to your environment by creating private subnets.
- **Security Groups** are like built-in firewalls that allow inbound and outbound connections to your network.
- **AWS Identity and Access Management (IAM)** allows specific control over access levels.
- **AWS Direct Connect** allows a dedicated network connection from a corporate network to AWS using industry-standard 802.1Q VLANs.
- **Amazon EBS Encryption** offers a simple and performant way to encrypt data at rest inside your disk volumes and data-in-transit between EC2 instances and EBS storage

Enterprise application security has three main components: network, access, and data. In the next sections, we'll look at how you can implement these capabilities in AWS and Tableau Server to enable a single report or dashboard to securely serve the needs of a broad and diverse user base, including both internal and external users.

Network

Network security for Tableau Server in AWS relies on the use of Amazon VPC network access control lists and security groups for traffic isolation. In addition SSL can and should be used for securing external communication.

Amazon VPC

An Amazon VPC is a distinct, isolated network within the cloud; network traffic within each Amazon VPC is isolated from all other Amazon VPCs. Using an Amazon VPC allows you to create your own network subnets and divide application layers into network subnets for a greater level of control.
We recommend that you install and run Tableau Server in one or more separate private subnets within your Amazon VPC so you can configure the network accordingly for access to Tableau Server and other data sets. Figure 1 shows a typical installation of a single-node Tableau Server in an Amazon VPC

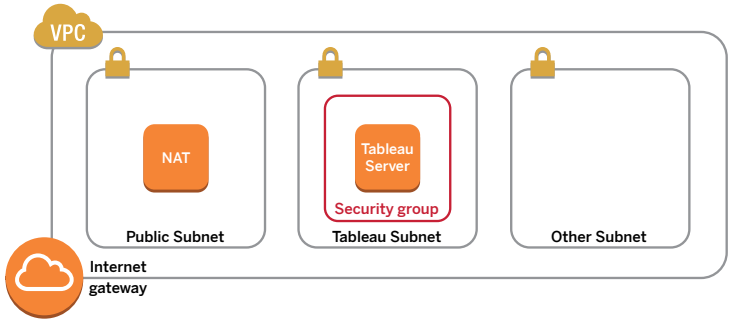


Figure 1: Amazon VPC Network Setup For a Single Node Deployment

For best practices related to VPC design, see Amazon Answers: VPC Design[4]

In Figure 1, incoming internet traffic cannot access Tableau Server directly since it is deployed to a private subnet. Instead, requests arrive through an Internet gateway and must transit a public subnet, then be routed to the private subnet and Tableau Server.

Individual subnets in an Amazon VPC can take advantage of network access control lists (nACL) as an additional layer of security for your VPC. The nACL acts as a firewall controlling traffic in and out of your subnets. By changing the default types of inbound and outbound traffic an nACL allows, and leveraging Security Groups, you can implement multiple layers of network access control.

If you choose to restrict certain types of traffic via an nACL, keep the following in mind:

- When deploying Tableau Server across multiple private subnets, Tableau processes in those subnets must be able to communicate. Ensure that you allow inbound and outbound traffic between private subnets for the ports listed under **All** and **Distributed/High Availability** categories in Tableau Server Help
- For the public-facing internet (Destination 0.0.0.0/0 or a smaller range you decide upon) you must enable outbound traffic for ephemeral ports plus 80/HTTP, 443/HTTPS and whatever ports are necessary to administer your Server – typically 3389/RDP.
- Network ACLs are stateless, meaning that responses to inbound traffic are subject to rules for outbound traffic.
- As a result, you should allow outbound responses between Tableau subnets and the public-facing internet on the ephemeral port range typically used for IPv4 responses: 1024-65535

Security Groups

You can define what types of network traffic can access the EC2 instances Tableau Server is installed on by using security groups. Amazon EC2 security groups act as a firewall that governs network traffic into and out of specific Amazon EC2 instances. You can define and assign security groups that are appropriate for your Amazon EC2 instances.

We recommend you create a new security group which is "customized" to protect your Tableau Server. All EC2 instances running Tableau Server should leverage this security group.

By default, EC2 instances are launched with security groups that are in an "allow nothing" state for inbound traffic, but you can make changes to allow the appropriate inbound traffic to the EC2 instance.

Here are the minimum requirements for connections to Tableau Server:

- On Windows, connection via RDP (port 3389) using a Remote Desktop client to access and manage the instance and services.
- Standard web traffic via HTTP (port 80) and HTTPS (port 443), to view content hosted on, and to publish to, Tableau Server
- Communication between Tableau Server components on different instances (if any) should be allowed. See the ports listed under **All** and **Distributed/High Availability** categories in Tableau Server Help

Based on these requirements, the security group for your Tableau Server deployment should enable only two standard ports for inbound traffic from the internet: HTTP 80, and HTTPS 443.

The security group should also allow private inbound Tableau-related traffic between nodes and/or subnets as defined in Tableau Server ports.

Note that Security Groups are *stateful*, therefore responses from Tableau to allowed inbound traffic will flow out regardless of outbound rules.

The Host (aka "Jump Server")

Consider using a bastion host to further secure your environment. Rather than allowing direct connections from the internet to Tableau Server via a Remote Desktop client, install a bastion EC2 instance in your VPC's public subnet.

The bastion computer should be associated with a distinct Security Group for the bastion which only allows inbound internet traffic on port 3389/RDP. The inbound security rules should also limit incoming RDP connections to a small list of IP addresses associated with trusted hosts in your organization. The Security Group for your Tableau Server will allow inbound connections on 3389/RDP from the bastion only.

Once an administrator has successfully logged into the bastion, (s)he then uses another RDP connection to "jump" to Tableau Server in its private subnet.

Creation and configuration of bastion hosts or Remote Desktop Gateways is beyond the scope of this whitepaper, but is demonstrated in the appendix

Client Access

Tableau recommends you employ SSL to protect client communication with Tableau Server.

By default, Tableau Server uses standard HTTP requests and responses. Tableau Server can be configured for HTTPS (SSL) with customer-supplied security certificates. When Tableau Server is configured for SSL, all content and communications between clients are encrypted and use the HTTPS protocol.

When you configure Tableau Server for SSL, the browser and SSL library on the server negotiate a common encryption level. Each web browser that accesses Tableau Server via SSL uses the standard SSL implementation provided by that browser.

> Tableau Server will listen for SSL traffic only on port 443. You may not configure custom ports for SSL/TLS

AWS Elastic Load Balancer (ELB) can also perform SSL termination on your behalf. Allowing ELB to handle (de)encryption of web traffic is an easy way to secure the client's connection with Tableau Server without needing to manually configure SSL on Tableau Server itself. For more information on this topic, see AWS Elastic Load Balancing: Support for SSL Termination[5].

Internal Communication

There are two aspects to communications between Tableau Server components in a distributed server installation: trust and transmission. Each server in a Tableau Server cluster uses a stringent trust model to ensure that it is receiving valid requests from other servers in the cluster. Trust is established by a whitelist of IP addresses, ports, and protocols. If any of these are invalid, the request is ignored.

Computers in the cluster that are running a gateway process accept requests from third parties (clients), unless a load balancer receives the requests. Servers that aren't running a gateway process accept requests only from other trusted members of the cluster. If you wish to encrypt network-based inter-process communication across multiple EC2 instances, consider implementing Internet Protocol Security (IPSec) at the operating system level.

Refer to the Tableau Server Administrator Guide for more information about configuring Tableau Server with SSL and enabling trust.

Access

Access security is used to establish the user's identity, to prevent unauthorized access, and to personalize each user's experience. Tableau server supports two authentication directories – local and Microsoft Active Directory. Any user who signs in and works with content in Tableau Server must have a user identity in the Tableau Server repository and must be assigned a site role

If the server is configured to use local authentication, the Tableau Server repository is used exclusively to authenticate the user. If the server is configured to use Active Directory authentication, then their username and password is verified using Active Directory.

Tableau Server supports several types of single-sign-on:

- **Security Assertion Markup Language (SAML)** – Tableau Server can be configured to use SAML for single sign-on (SSO). In this model, an external identity provider (IdP) authenticates the user's credentials, and then sends a security assertion to Tableau Server with information about the user's identity.
- **OpenID Connect** – A simple identity layer on top of the OAuth 2.0 protocol allows clients to verify the identity of the end-user based on the authentication performed by an Authorization Server, as well as to obtain basic profile information about the end-user in an interoperable and REST-like manner.
- **Kerberos** – If Tableau Server is configured to use Windows Active Directory authentication and Kerberos is enabled, users can gain access to Tableau Server based on their Windows identities.

- **Trusted authentication** – In trusted authentication, you set up a trusted relationship between Tableau Server and one or more web servers or web services. When Tableau Server receives requests with a redeemable token or ticket issued to a trusted web server/service, it is deemed valid.

Tableau Server also provides a personalized experience for users by creating an account for each named user of the system. For more information, see the Authentication section of the Tableau Server Administrator Guide.

AWS Directory Service

AWS Directory Service is a managed service that allows you to connect your AWS resources to an existing on-premises directory such as Microsoft Active Directory (with AD Connector), or to set up a new, stand-alone directory in the AWS cloud (with Simple AD). Connecting to an on-premises directory is easy, and once this connection is established, all users can access AWS resources and applications with their existing corporate credentials.

Using the AWS Directory Service, you can choose to use Active Directory-based authentication instead of local authentication. (Local authentication creates users and assigns passwords using Tableau Server's built-in user management system.) To set up Active Directory-based authentication, in the configuration step after installing Tableau Server, you must choose Active Directory. It is not possible to switch between Active Directory and local authentication later.

The Active Directory authentication model uses the Microsoft Security Support Provider Interface (SSPI) to sign in your users automatically, based on their Windows user name and password. This creates an experience similar to single sign-on (SSO).

Data

Tableau Server uses native drivers (relying on a generic ODBC adapter when native drivers are not available) to connect to databases whenever possible, for processing result sets, for refreshing extracts, and for all other communications with the database. You can configure the driver to communicate on non-standard ports or use transport encryption, but this type of configuration is transparent to Tableau Server.

Since the Tableau Server-to-database communication is typically behind a firewall, most customers choose not to encrypt this communication. If you connect to remote data sources over an untrusted network, consider the use of an AWS hardware Virtual Private Network or a software-based VPN appliance.

If you wish, you can configure encrypted communication between Tableau Server's metadata repository (PostgresSQL) and Tableau Server components. See Configure Internal SSL for more information .

Connecting to Data Stores in AWS

You can launch AWS resources, such as Amazon Relational Database Service (Amazon RDS), Amazon Elastic MapReduce (Amazon EMR), or Amazon Redshift, into an Amazon VPC. By placing the Tableau Server into the same Amazon VPC as your data stores, you can ensure that your traffic never leaves the Amazon VPC. For more information on using Tableau and Redshift together, visit our resource page.

You can use subnets with security groups to launch your resources into different layers but allow them to communicate securely within an Amazon VPC, as illustrated in Figure 2.

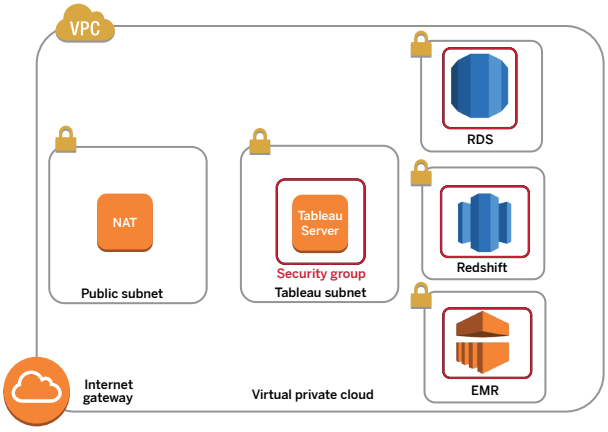


Figure 2: Amazon VPC Network Setup with Data Stores

Connecting to Data Stores Outside of AWS

You can optionally connect your Amazon VPC to your own corporate data center by using an IPsec hardware VPN connection, thus making the AWS cloud an extension of your data center. A VPN connection consists of a virtual private gateway attached to your Amazon VPC and a customer gateway located in your data center.

AWS Direct Connect is a network service that provides an alternative to using the Internet to utilize AWS cloud services. AWS Direct Connect lets you establish a dedicated network connection by using industry-standard 802.1Q VLANs. You can use the same connection to access public resources (such as objects stored in Amazon Simple Storage Service [Amazon S3] using public IP address space) and private resources (such as Amazon EC2 instances running within an Amazon VPC using private IP space), while maintaining network separation between the public and private environments.

Encrypting Data at Rest

Amazon EBS encryption offers a transparent and simple way to encrypt volumes which may contain personally identifiable information (PII).

EBS encryption encrypts both data at rest inside the volume and data in transit between the volume and the instance using AES-256. Internal testing by Tableau shows that that utilizing this feature has little-to-no impact on Tableau Server performance. We therefore recommend you take advantage of this service regardless of whether your systems store PII or not.

Scaling Tableau Server on AWS

Tableau Server is architected to scale up with more CPU cores and memory, and scale out when you add servers. This architecture allows you to maximize the use of compute resources while giving you the ability to scale massively.

With a single-server deployment on AWS, all Tableau Server processes run on a single instance, as shown in Figure 3. This is the most basic configuration for Tableau Server.

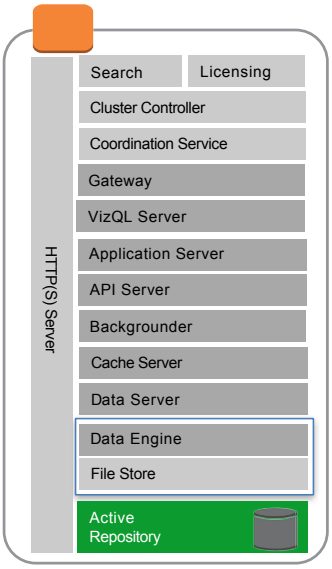


Figure 3: Tableau Server Components on a Single Amazon EC2 Instance

To take advantage of additional resiliency gained when running Tableau Server across multiple AWS Availability Zones (AZs), we recommend you consider scaling out before you scale up. Add additional EC2 instances, install Tableau workers on the instances, and run new copies of Tableau services on them.

> Although you can split an 8-core license on two 4-core (8 vCPU on EC2) machines, we recommend that you scale in increments of 8-core or more (16 vCPU) machines only.

High Availability

To configure a cluster that provides failover support for the data engine and repository processes, you need at least three EC2 instances: one will act as a primary for your Tableau Server while the other two instances act as Tableau workers.

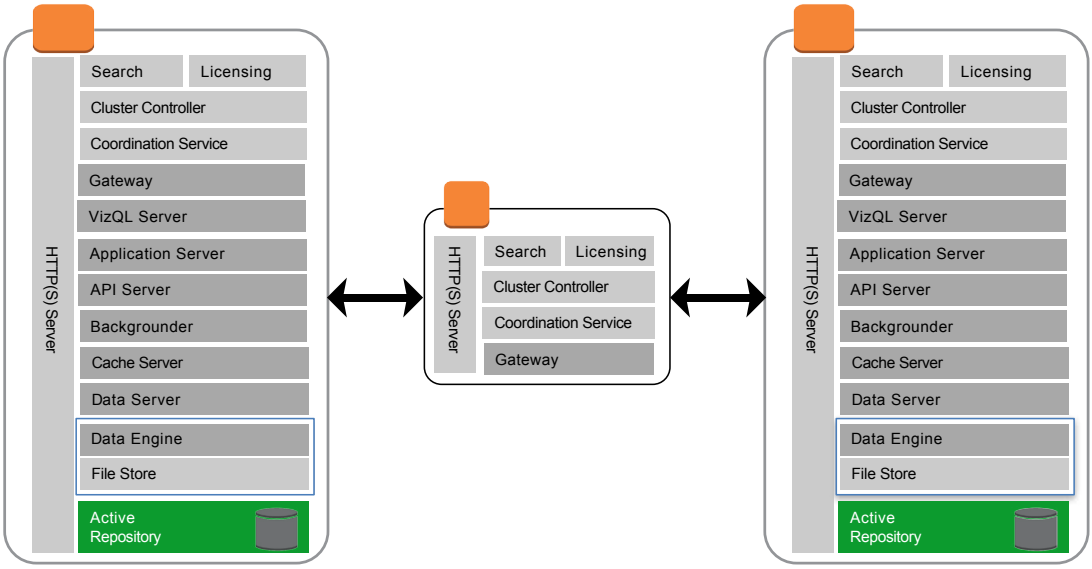


Figure 4: Tableau Server in Failover Mode

An ideally configured distributed Tableau Server environment status page should look similar to Figure 5.

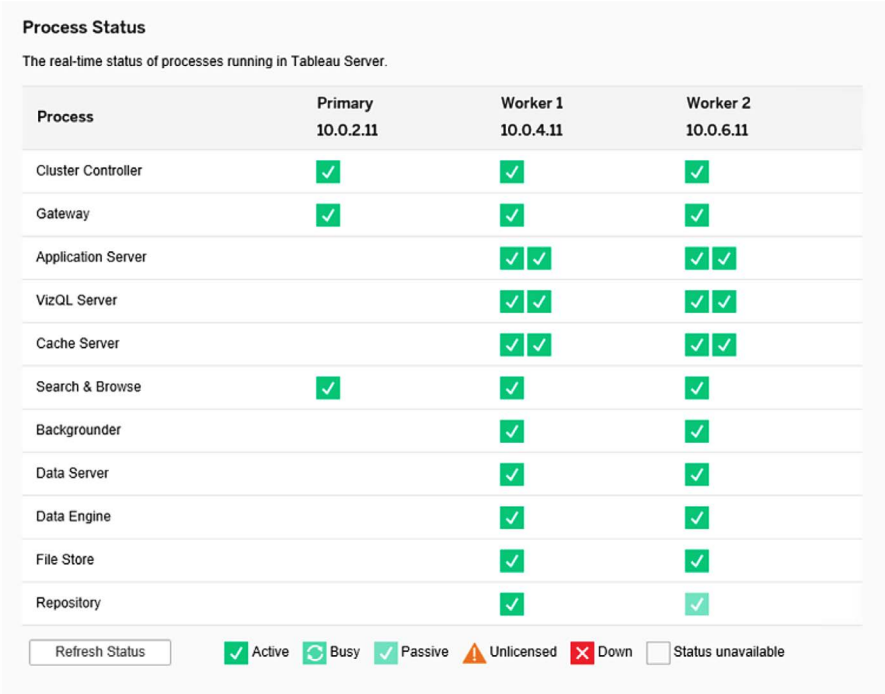


Figure 5: Status Page of Tableau Server in Failover Mode

In this deployment, the data engine and repository processes have been moved from the primary to a worker, and the primary is running only the gateway process along with Search & Browse and Cluster Controller. Furthermore, only the repository component is in a passive state, and is kept up to date using replication; all other components are active-active. In the event of a failover, the passive repository becomes active, and your deployment continues to function.

Load Balancing

On AWS, **Elastic Load Balancing (ELB)** automatically distributes incoming application traffic across multiple Amazon EC2 instances in the cloud. It enables you to achieve greater levels of fault tolerance in your applications, and seamlessly provides the required amount of load balancing capacity needed to distribute application traffic.

You can and should use Elastic Load Balancing to distribute requests across multiple gateways in a Tableau Server cluster. Figures 4 and 5 show that all three nodes have gateways, which are used to route requests to available server processes. Unlike the repository process, there aren't any passive or standby gateway processes—all gateways are active. When you add a load balancer to a Tableau Server cluster, the load balancer becomes the first point of entry to the cluster. Users will call the load balancer URL which then communicates with the cluster.

> **Note:** When running Tableau Server across multiple Availability Zones, enable **cross-zone load balancing** on your ELB to evenly distribute incoming requests all across all EC2 instances in your AZs. More information can be found here.

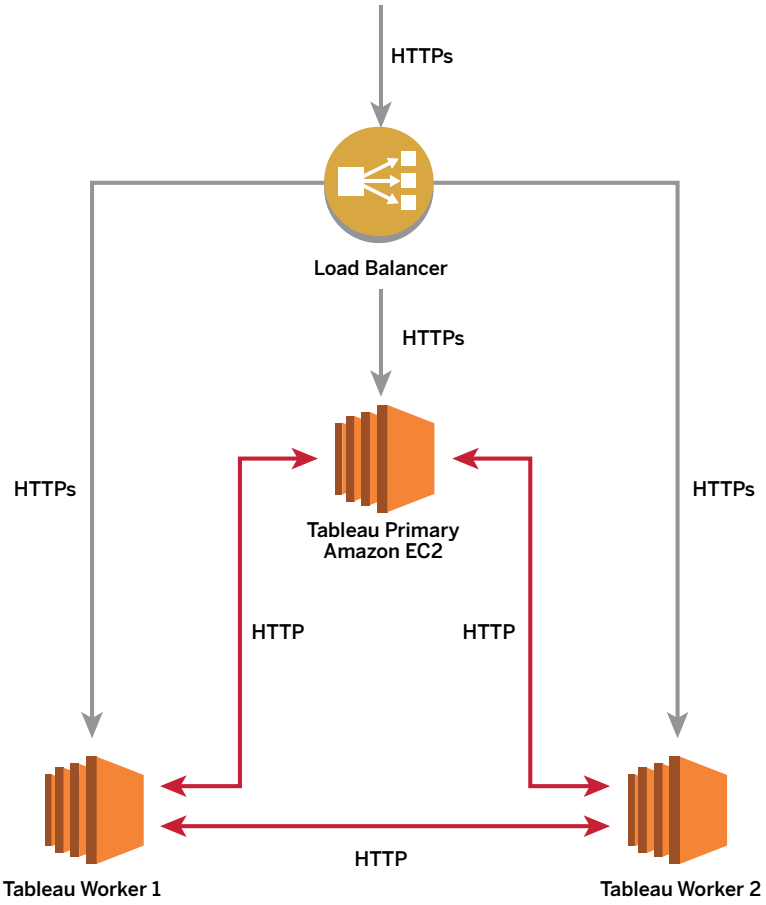


Figure 6: Tableau Server with Elastic Load Balancing

Once you have provided redundancy for the data engine, repository, and gateway by adding multiple workers, you can add additional redundancy for the Primary Tableau Server. You can do this by creating a backup of the primary Tableau Server. Although the backup primary needs to be licensed during installation, it does not count as one of the three environments allowable under the Tableau EULA.

To configure for high availability, you need to run a failover cluster plus an additional computer as the backup primary for your primary Tableau Server. If you configure for high availability, the primary Tableau Server and the backup primary may be running few or no Tableau Server processes.

When a licensed process starts or restarts, the process checks with the Tableau Server License Manager service on the primary node to verify there is a valid license. When the License Manager validates the license, the process is fully functional and able to respond to requests from other Tableau Server processes. Once a licensed process has received confirmation from the License Manager, the process does not need to reconfirm the license for 72 hours, or until the process restarts. If the process is not able to verify that it is licensed (if the primary node is unavailable, for example) it cannot run, but it continues to check for a valid license until it confirms the license.

Conclusion

Deployment of Tableau Server on AWS differs very little from deployment on traditional infrastructure. In this whitepaper, we presented a number of AWS platform considerations relating to security, storage, compute configurations, management, and monitoring that need to be considered to get the best out of your Tableau Server deployment on AWS. Following the best practices and guidelines provided in this paper will help you achieve optimal performance, availability, and reliability.

Contributors

The following individuals and organizations contributed to this document:

- Rahul Bhartia, Product Manager, Amazon Web Services
- Russell Christopher, Product Manager, Tableau Software
- Ken Chestnut, Global Ecosystem Lead, Amazon Web Services
- Ashley Kramer, Director of Product Management, Tableau Software
- Jazmyn Li, Sales Consultant, Tableau Software
- Yu Hua Lim, Solutions Architect, Amazon Web Services

Appendix

Deploy a Tableau Server cluster on AWS: Step-By-Step

In this section, we'll walk through the steps of deploying a Tableau Server cluster on AWS. We'll also show how to use some of the services and Tableau Server features mentioned in previous sections for deploying Tableau Server in a highly available and scalable configuration on AWS.

The steps below model the output of the Tableau Server on AWS Quickstart and demonstrate best practices for deployment on AWS.

Create a VPC Architected for High Availability

The following steps assume that you've created an Amazon VPC in the AWS cloud with at least six subnets (three public and three private) in three different Availability Zones, as illustrated in Figure 7.

> Before following the deployment steps below, ensure that you have access to an AWS account, Tableau Server installation files, and a valid product key that will enable the use of a minimum of 16 cores for production.
>
> You should review the entire appendix before you begin creating resources in AWS. Make sure you understand whether you will leverage an internal or external load balancer, and don't create subnets that are too large.

If you intend to utilize an AWS Elastic Load Balancer (ELB), create relatively small subnets with a /27 netmask. More details on subnet size may be found in the topic Create a Load Balancer as a Front-End for the Tableau Server Cluster on page 25.

For information about how to create a VPC with public and private subnets, see Scenario 2: VPC with Public and Private Subnets in the Amazon VPC User Guide.

> The deployment pattern demonstrated here is unique to AWS due to how Availability Zones (AZ) are designed. An AZ provides inexpensive, low-latency direct network connectivity to other AZs in the same region. Deploying Tableau Server across multiple subnets as suggested below is not supported outside of AWS.

Create Bastions and NAT Gateways

Three EC2 instances will act as bastion hosts for Tableau Server instances within the private subnets. After you've created the VPC, either:

- Launch an EC2 instance running Windows in each of the three public subnets. You will login to these bastions via RDP and then "manually jump" to your private subnets via a different RDP connection initiated from the bastion
- Deploy a Remote Desktop Gateway. This approach is more complex to deploy, but makes connecting to your private subnets more seamless. For information on how to launch and connect to a Windows instance, see Getting Started with Amazon EC2 Windows Instances in the Amazon EC2 User Guide. For further information on how to configure Remote Desktop gateways, see Controlling Network Access to EC2 Instances Using a Bastion Server in the AWS Blog

Setting up NAT Gateways in the public subnets allows Tableau Server instances in the private subnets to connect to the Internet, but prevents the internet from initiating a direct connection with Tableau Server instances in the private subnets.

Setup a NAT Gateway in each of your three public subnets. For further information on how to configure NAT Gateways, see NAT Gateways in the Amazon VPC User Guide .

After completing the steps above, your VPC should resemble Figure 7.

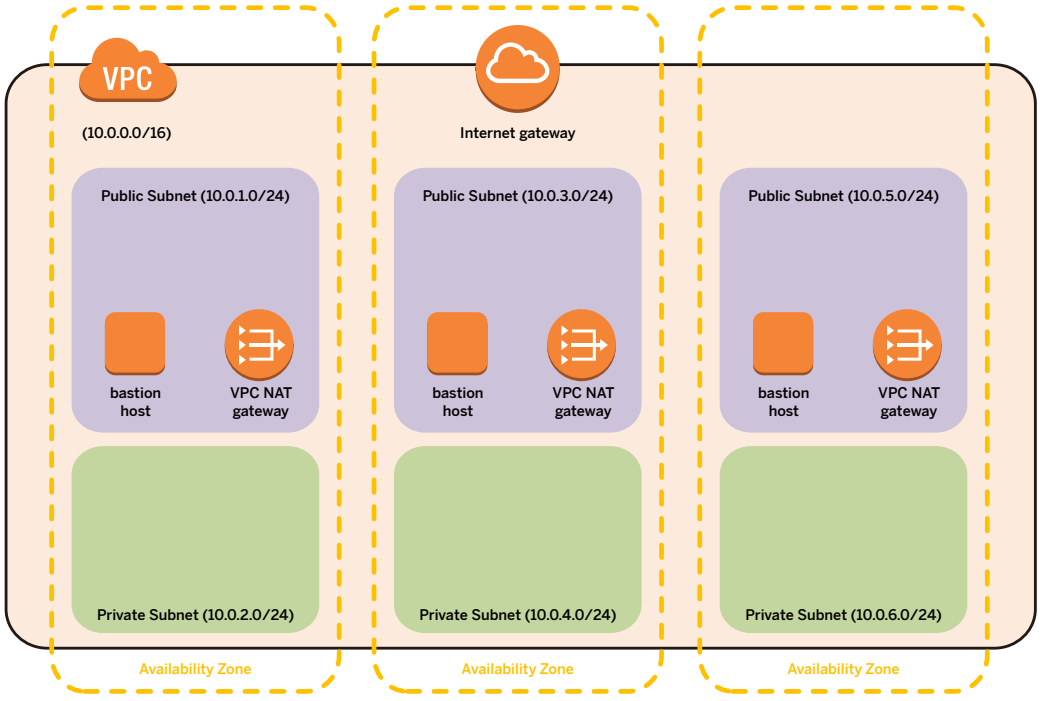


Figure 7: AWS VPC with private/public subnets and NAT gateways

Deploy Three EC2 Instances in the VPC

Next, deploy one AWS EC2 instance (m4.4xlarge or r4.4xlarge with the latest Windows Server 2016 Base AMI) in each of your three Availability Zones, as illustrated in Figure 8.

One node will be utilized as the primary Tableau Server node and the other two nodes as workers. If you wish, the EC2 instances can be auto-joined to an AD domain by following the steps in Joining a Windows Instance to an AWS Directory Service Domain in the *AWS documentation.*

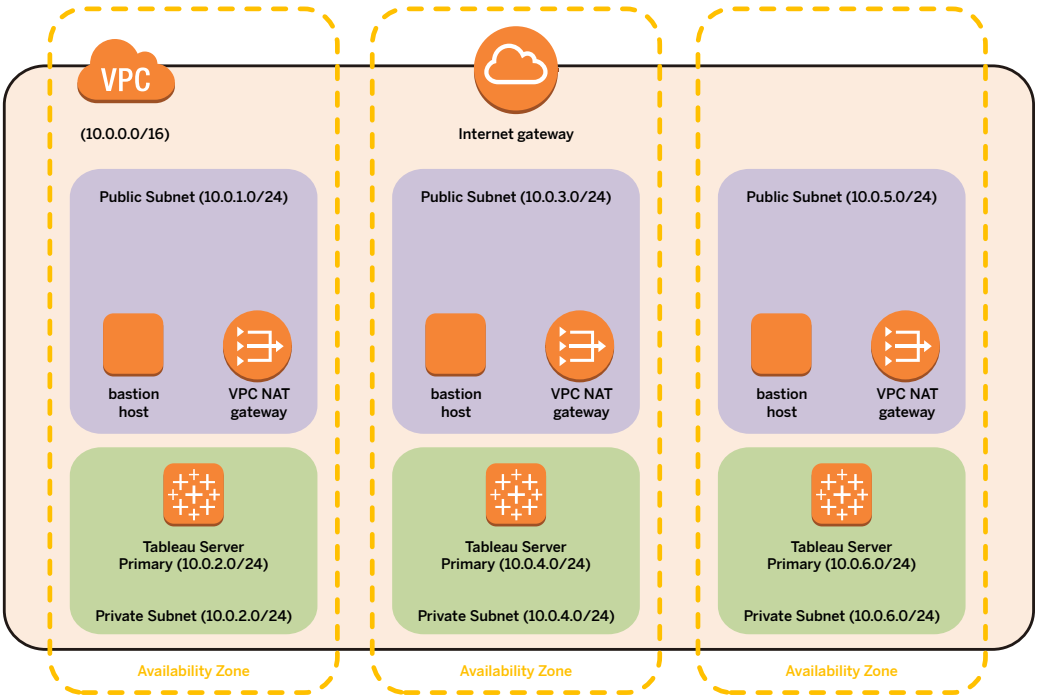


Figure 8: Tableau Server deployed in a highly available fashion

After launching the new EC2 instances, connect to them via RDP from one of the bastion instances.

> If the Tableau Server's Primary node fails, Tableau's licensing service will stop responding to requests from Tableau processes which could eventually cause them to de-activate. To protect against this event, set up a **Backup Primary** server in a different Availability Zone than the current Primary. For further information on how to create and failover to the Back Up Primary Server see Use a Backup Primary in the *Tableau Server online help* .

Install and Configure Tableau Server

Next, you will install Tableau Server on the EC2 instances you deployed and configure the Tableau Server nodes as primary and worker servers.

Install the Tableau Server primary.

To avoid disk contention, install Tableau Server on its own disk drive.

On the Amazon EC2 instance you have chosen to act as Primary, follow these steps:

- Download the Tableau Server installation file from the Tableau Alternate Downloads Site. Make sure to download the Primary Networked Server.
- Run the Tableau Server installation file, and follow the on-screen instructions to complete setup and install the application.
- After the installation is complete, choose **Next** to open the **Product Key Manager** window and enter the product keys you obtained from the Tableau Customer Portal.
- The **Tableau Server Configuration Utility** will be displayed. You can set some configuration options now, before the server starts at the end of the installation process. After installation completes, more options will become available.
- Configure options on the **General** tab:

a. In **Server Run As User**, specify a domain account and password if your plan to leverage Active Directory Authentication for your users. Otherwise, leave the default value of NT AUTHORITY\NetworkService in place.

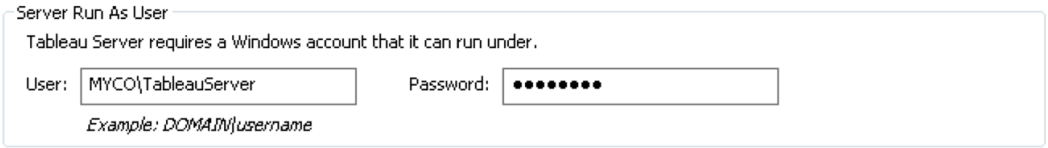


Figure 9: Choosing a Domain Account for Tableau Server

b. Choose Use **Local Authentication** or **Active Directory** to authenticate users on the server. If you select Active Directory, you must also specify the **Domain**.

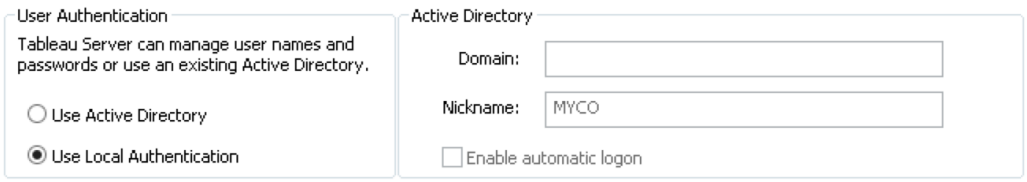


Figure 10: Configuring Tableau Server to Use Local Authentication

If you choose to protect client connections to Tableau Server with SSL, you may use ELB SSL Termination or configure the cluster itself to leverage SSL. Refer to instructions in the Tableau Server Administrator Guide:

- External SSL for clients

Using SSL ensures that your client communications with Tableau Server are encrypted. You should use SSL when Tableau Server is accessible from the Internet.

Stop Tableau Server on the primary node (see Tableau Server Monitor in the Tableau Server Administrator Guide to learn how).

Install Tableau Server workers

On the Amazon EC2 instances tagged as Worker1 and Worker2, follow these steps:

- Download the Tableau Server Worker software from the Tableau Customer Account Center.
- Run Tableau Server Worker Setup on all additional computers that you want to add to the Tableau Server cluster.
- During installation, you will be asked to provide information about the primary server. Specify the **internal DNS hostname** of the primary or its **private IP address**.

Configure Tableau Server in distributed mode

Once the worker software is installed on worker computers, return to the primary server and open the Tableau Server Configuration Utility found in the Tableau Server program group.

- On the **Servers** tab, click **Add** to add a worker server. Enter its private IPv4 address or internal computer name. Enter appropriate values for each Process as defined by Tableau's Performance Tuning Examples . Check the **Repository**, **Gateway**, and **Search** & **Browse** check boxes. Click **OK**.

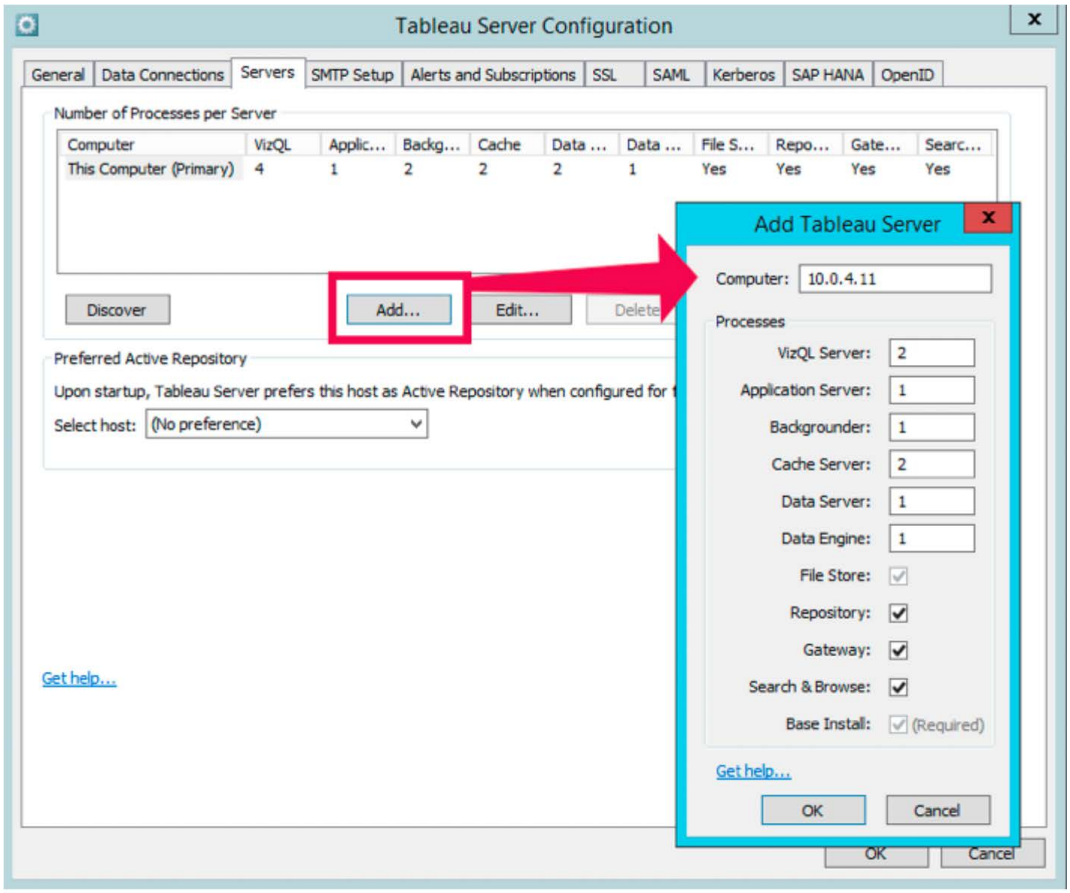


Figure 11: Setting up a Worker for High Availability Deployment

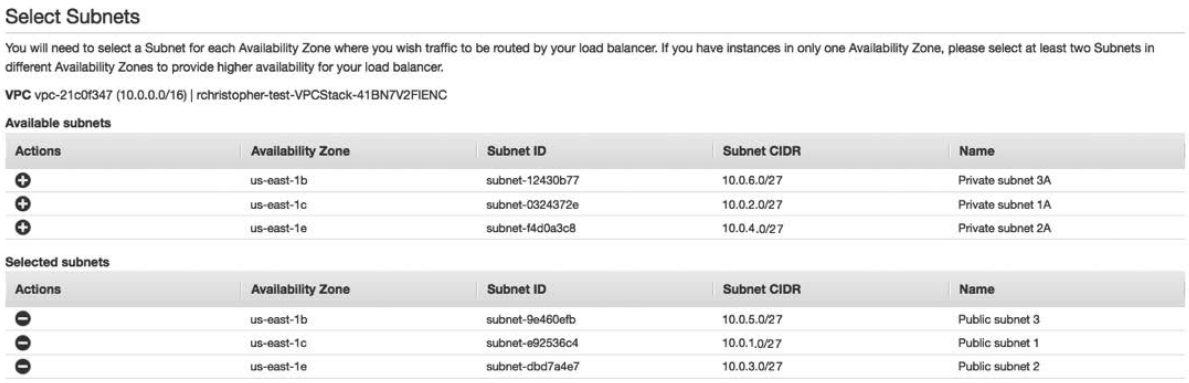
- Choose **Add** to add a second worker server. Enter its private IPv4 address or internal computer name. Enter 1 for every process except for **Data Engine** (set that to **0**). Leave the **Repository** check box cleared, but select the Gateway check box. Choose **OK**.
- Click **OK** in the **Tableau Server Configuration** window. After a few moments you will be prompted to restart Tableau Server. Do so.
- After the Server starts, wait for a few minutes to ensure a copy of the repository is copied to Worker1. You may monitor this process via Tableau Server's Status page.
- Stop Tableau Server and Launch the **Tableau Server Configuration** Utility once more.
- Click the **Servers** tab, select **This Computer (Primary)** and choose **Edit**. Set every process to 0, clear the **Repository** check box, but keep **Gateway** selected. Click **OK**.
- Select the second worker and choose **Edit**. Set **Data Engine** to **1** and check the **Repository** check box. Choose **OK**, and then **OK** again to close the Configuration Utility. Start Tableau Server.

Create a Load Balancer as a Front-End for the Tableau Server Cluster

Follow the seven steps outlined in **Getting Started with Elastic Load Balancing** in the AWS documentation to launch a **Classic** load balancer within your VPC.

- If you need an ELB accessible from outside AWS, it should include the three **public** subnets in your VPC. Otherwise, check the **Create an internal load balancer** box and choose all three private subnets. Figure 12 shows three public subnets being selected for the load balancer which will service external traffic.

If you choose to expose the load balancer with a public endpoint, we recommend that you configure Elastic Load Balancing with SSL. See the AWS documentation topic Create an HTTPS Load Balancer for more information.



Select Subnets

You will need to select a Subnet for each Availability Zone where you wish traffic to be routed by your load balancer. If you have instances in only one Availability Zone, please select at least two Subnets in different Availability Zones to provide higher availability for your load balancer.

VPC vpc-21c0f347 (10.0.0.0/16) | rchristopher-test-VPCStack-41BN7V2FIENC

Available subnets

Actions	Availability Zone	Subnet ID	Subnet CIDR	Name
⊕	us-east-1b	subnet-12430b77	10.0.6.0/27	Private subnet 3A
⊕	us-east-1c	subnet-0324372e	10.0.2.0/27	Private subnet 1A
⊕	us-east-1e	subnet-f4d0a3c8	10.0.4.0/27	Private subnet 2A

Selected subnets

Actions	Availability Zone	Subnet ID	Subnet CIDR	Name
⊖	us-east-1b	subnet-9e460efb	10.0.5.0/27	Public subnet 3
⊖	us-east-1c	subnet-e92536c4	10.0.1.0/27	Public subnet 1
⊖	us-east-1e	subnet-dbd7a4e7	10.0.3.0/27	Public subnet 2

Figure 12: Load Balancer Setup with Public Subnets

- In step two (Assign Security Groups to Your Load Balancer) of Getting Started with Elastic Load Balancing, ensure that your security group is configured to allow access on port 80 or 443 only, with source limited to hosts or ranges of hosts that will access Tableau Server.
- In step 4 (Configure Health Checks), you can specify the ping path as "/".
- In step 5 (Register Ec2 Instances with Your Load Balancer) , select the Tableau Server instances and ensure that **Enable Cross-Zone Load Balancing** is selected so that the load balancer can load-balance the traffic across the instances in multiple Availability Zones.

> When you deploy Tableau Server with Elastic Load Balancing, Additional ELB-related information must be added to the Tableau Server configuration. If you forget these steps, some URLs displayed by Tableau in "Share" and embedded in subscription emails may be incorrect.

In order to minimize the number of ELB-related IP addresses Tableau must manage and trust, subnets (whether public or private) in which you place your ELB should be small.

Create these subnets with the smallest possible CIDR block, and then configure the Tableau Server with all the IP addresses for each of the three CIDR ranges. The following example demonstrates how to implement same.

Scenario: You wish to deploy an internet-facing ELB which will redirect traffic to Tableau Server. Tableau Server will be deployed to three private subnets, each of which resides in a different AZ

- Create three public subnets for your load balancer to match the three private subnets Tableau is deployed to. Each subnet should have a CIDR block with a /27 netmask, generating a 32 IP address subnet.
- Create three private subnets for Tableau and deploy the software.

In the Tableau Server \bin directory, enter the following command, where the value in quotes is the public hostname or IP address that will be used to reach Tableau Server through the load balancer. Examples of this values might be the DNS Hostname of the ELB itself, or an A record in Route 53 which points to the ELB:

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

AWS reserves the first four and last one IP address of each subnet you create for internal use. Therefore, each of your public subnets will contain a total of 27 "usable" addresses. At any given time, ELB may claim one or more of these addresses.

Enter the following command, providing the fifth through thirty-first IP addresses for all subnets used by Elastic Load Balancing. Given example subnets of 10.0.1.0/27, 10.0.3.0/27, and 10.0.5.0/27, the complete list should contain 81 addresses and look like this:

```
tabadmin set gateway.trusted "10.0.1.4, ...,10.0.1.31,10.0.3.4,...,
10.0.3.31,10.0.5.4,..., 10.0.5.31"
```

Run the config command:

```
tabadmin config
```

Start the server so the changes can take effect. After your Tableau Server workers are registered as "InService" by AWS Elastic Load Balancer, you may access Tableau through the DNS name of the ELB itself.

Using the AWS Marketplace

Another way to get started is to use the Tableau Server Amazon Machine Image (AMI), which is available in the AWS Marketplace. The Tableau Server AMI is pre-packaged and configured for deployment in a typical environment on AWS.

There are two variants of the Marketplace offering:

- **Bring your own license (BYOL):** A pre-configured AMI which you add your Tableau Server license to after it deployed. You are charged an hourly fee by AWS for the use of their resources.
- **Hourly:** An AMI in which the Tableau Server license fee is "included" in the cost of running the instance. You are charged an hourly fee by AWS which includes the cost of their resources and a Tableau Server license for the number of users you wish to provision

You should be aware of the following limitations of Marketplace images:

BYOL Limitations

BYOL images are configured to use local authentication. If you wish to use Active Directory instead, you must do a complete uninstall of Tableau Server and then re-install using a the new authentication style

Tableau Server is configured to run as a single-node Primary on BYOL images. If you wish to use the instance as a worker, you must remove the pre-installed Primary, then download and setup worker software manually.

Hourly Limitations

Hourly images are also configured to use local authentication. Unlike BYOL, you may not uninstall and re-install the product. Therefore, if you wish to leverage AD authentication, you should avoid this variant.

Hourly AMIs may not be upgraded in-place. If you wish to upgrade your server, you must:

- Backup your existing deployment with the tabadmin backup command
- Launch a new Hourly AMI and restore the backup to it using tabadmin restore --no-config
- Retire the original Hourly instance

The Hourly AMI can be deployed as a single-node instance only.

Utilizing Tableau Server on BYOL

To deploy Tableau Server on a BYOL image, follow these steps:

- In the AWS Marketplace, search for **Tableau Server BYOL**.
- Find **Tableau Server (BYOL)** and click **Select**. If prompted, sign in to your AWS account.
- If you have never launched the Tableau Server AMI, you will be presented with product details. Review them, and then choose **Continue** when you are ready to deploy Tableau Server on AWS.
- On the AMI details page, choose the **1-Click Launch** tab.

You can configure all the settings for your instance on this tab. This option allows you to launch only one Amazon EC2 instance at a time.

> **Note:** If you prefer to use the Amazon EC2 console to launch your instance, click the **Manual Launch** tab from the AMI details page. This option allows you to launch multiple Amazon EC2 instances.

- In the **Region** section, select the AWS region where you want to launch your Tableau Server instance.
- In the **EC2 Instance Type** section, choose your preferred instance type. See the previous section for recommendations.
- In the **VPC Settings** section, choose the Amazon Virtual Private Cloud (Amazon VPC) that your instance will be launched in.
- In the **Security Group** section, review the proposed security group settings and either choose from your existing security groups or accept the new proposed security group.
- In the **Key Pair** section, choose a key pair to associate with the instance. This key is used to connect to your instance over Secure Shell (SSH).
- Choose **Accept Terms & Launch with 1-Click** to create your new instance.

For step-by-step information about using Tableau Server after launching it successfully from the AWS Marketplace, see:

AWS Marketplace — Tableau Server in the Amazon Web Services Cloud

Amazon EBS Volume Types

Getting Started with Tableau Server

AWS Quick Starts — Tableau Server on AWS

AWS Single VPC Design

AWS Elastic Load Balancing: Support for SSL Termination

Tableau Server Administrator Overview

Tableau Server Authentication

Configuring Internal SSL

Scenario 2: VPC with Public and Private Subnets (NAT)

Getting Started with Amazon EC2 Windows Instances

Network to access to EC2 using a Bastion Server

NAT Gateways

Joining a Windows Instance to an AWS Directory Service Domain

Using a Backup Primary

Tableau Alternate Downloads Site

AWS Elastic Load Balancing: Support for SSL Termination

Tableau Server Monitor

Tableau Performance Tuning Examples

Creating a Classic Load Balancer with an HTTPS Listener

About Tableau

Tableau helps people transform data into actionable insights that make an impact. Easily connect to data stored anywhere, in any format. Quickly perform ad hoc analyses that reveal hidden opportunities. Drag and drop to create interactive dashboards with advanced visual analytics. Then share across your organization and empower teammates to explore their perspective on data. From global enterprises to early-stage startups and small businesses, people everywhere use Tableau's analytics platform to see and understand their data.