



Next Generation Cloud BI:
**Tableau Server
on Microsoft Azure**

Chris Bullock, Senior Sales Consultant
Madeleine Cornelius, Product Manager

Contents

Introduction	3
Start here	3
Tableau Server on Microsoft Azure: an overview	3
Purpose	3
Tableau Server on Microsoft Azure: an overview	4
A pattern for enterprise deployment.....	5
Virtual machine selection and performance guidance	7
High availability strategy	8
Enterprise DMZ strategy and hub-and-spoke considerations	10
Load balancing strategy	11
Identity and access management	12
Choosing an identity store	12
Azure Active Directory & Tableau Server	14
Active Directory hybrid identity & Tableau Server	15
Data source authentication and access	16
Data environment	17
Data sources	17
Data governance and management with Tableau	17
Planning for operational success	18
Cost management	18
Data transfer costs	19
Automation	20
ExpressRoute	20

Contents (Con't)

Deployment steps	20
Create the VNet, subnets, and security group	21
Create the jumpbox	23
Create the first Tableau node	26
Create additional nodes as needed	36
Create a load balancer	37
(Optional) Peer the hub and spoke and apply firewall rules	45
Tableau Server on Azure HA reference architecture	48
About Tableau and additional resources	49

Introduction

Start here

This whitepaper offers a holistic overview of the resources and techniques required for deploying Tableau Server on Microsoft Azure. Since your deployment will be unique to your organization, this paper is intended to help inform your deployment decisions. In addition to explaining step-by-step instructions for setting up your environment, we explain the theory behind the services and best practice recommendations.

For the simplest deployment of a single server, proceed directly to the deployment steps section.

We recommend estimating costs using this [Azure pricing calculator](#).

Tableau Server on Microsoft Azure: an overview

Organizations today recognize that modern analytics at scale need a self-service approach that empowers business users of all analytical skill levels. Tableau is uniquely positioned to deliver this experience, as we consistently raise the bar on what you can expect from your BI investment—from a seamless user experience, deep analytical capabilities, end-to-end data governance, and flexible integrations and extensibility. Our platform is simple to deploy and scale, and as intuitive to learn as it is powerful—helping individuals and organizations get insights and value fast. Everything we do is driven by our mission to help people see and understand data, which is demonstrated by our commitment to business intelligence and analytics, relentless customer-focused innovations, and our global community—unrivaled in creativity, diversity, and passion.

Year after year the public cloud gains traction and continues to increase in adoption. Not only do the majority of organizations running workloads on the public cloud, most are leveraging multiple public clouds. According to the [RightScale 2019 State of the Cloud Report from Flexera](#), 91% of enterprises have adopted the public cloud, with 84% of enterprises having a multi-cloud strategy. The same report also highlights Azure as one of the fastest growing public clouds, especially for enterprises. [Microsoft's 2019 Q4 earnings](#) showed that “revenue from Azure increased 64% year over year, the lowest growth rate in at least four years.” The hyper-growth of this industry means that organizations must be able to deploy cloud workloads with confidence while also maintaining flexibility.

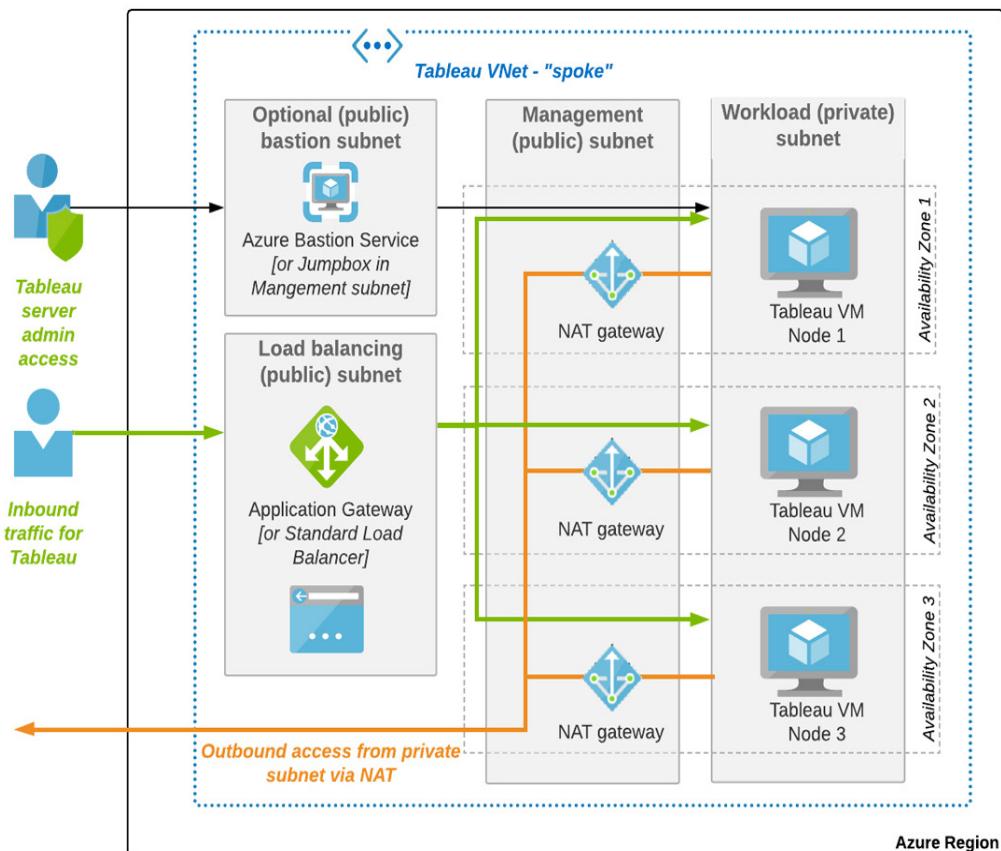
Microsoft Azure provides a flexible, cost-effective, and high-performance cloud computing platform to deploy, scale, and manage your Tableau Server implementation. With Azure Virtual Machines, you can spin up instances as you need, when you need them—there are no upfront investments, hardware procurement, or physical deployment required. Together, Tableau and Azure provide a comprehensive business intelligence solution that can be implemented quickly, secured easily, scaled efficiently, and used by everyone.

A pattern for enterprise deployment

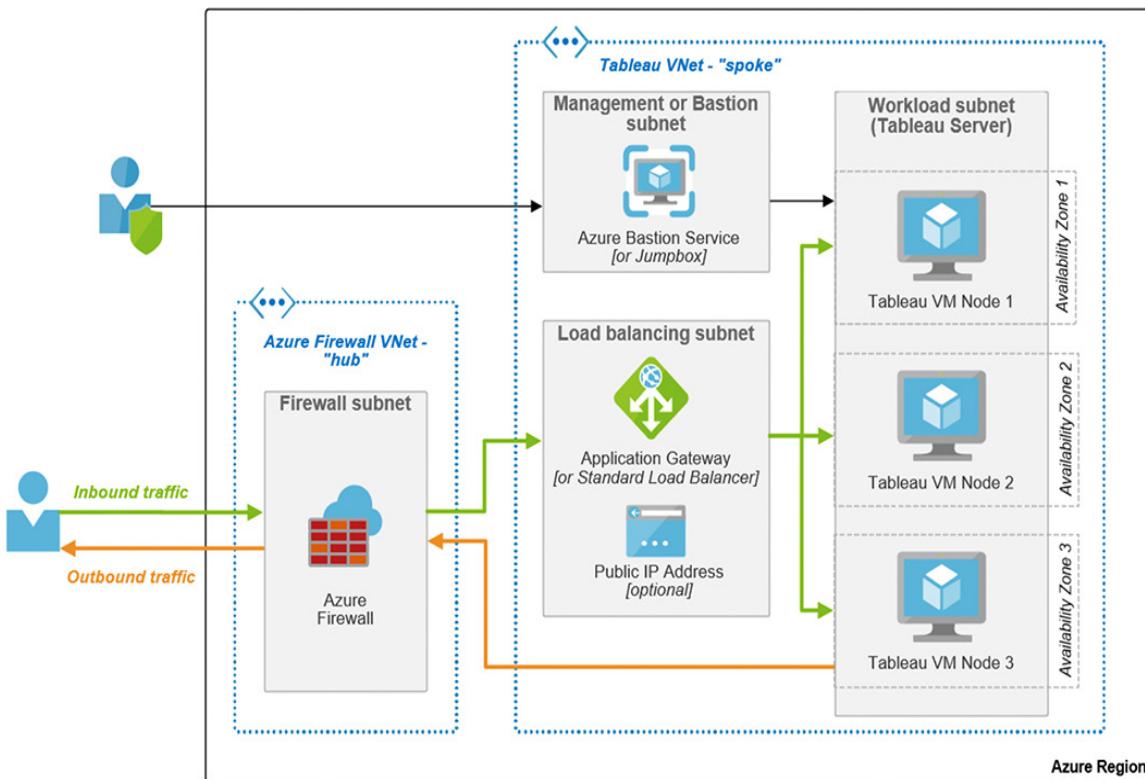
Tableau Server can be deployed in many configurations on all major public clouds. This section will explore best practices for the elements of enterprise architecture, and provide a typical high availability (HA) configuration and a hub and spoke version for Tableau Server on Microsoft Azure.

For the simplest deployment of a single server, proceed directly to the deployment steps section.

The diagram below shows the Tableau HA pattern used in all public clouds, tailored for Azure.



The second diagram is entirely optional and elaborates on this same pattern for deployment as a hub and spoke architecture. Coordination from hub to spoke is provided by Azure Firewall or a similar service.



Virtual machine selection and performance guidance

One of the early decisions in planning your Tableau deployment is machine sizing and selection. The size of your user base, how they will interact with Tableau content, and your data environment are all key considerations. We provide resources to help guide you through this decision process, including a whitepaper discussing details of [Tableau Server sizing and performance](#).

Sizing your deployment on the cloud is similar to sizing an on-premises deployment. Azure offers a variety of [virtual machine \(VM\) sizes](#) to accommodate your specific needs. These machines offer a broad range of CPU, RAM, and Storage allocation. For your first Tableau Server deployment you should follow our [minimum recommended hardware](#). The cloud equivalent of CPUs are referred to as virtual CPUs (vCPUs) and follow a 2:1 ratio. Therefore the [Standard_D16s_v3](#) provides the equivalent of 8 CPUs (16 vCPUs), 64 GiB memory and 128 GiB SSD storage. If you are looking to trial Tableau Server or want a baseline hardware recommendation to use for testing your usage patterns this is a great size to start with.

As your Tableau usage grows—either by adding users, content, data sources, or all of the above—you may want to adjust the underlying hardware to maintain optimal performance. Deploying on the cloud makes it easy to adjust your hardware to suit the needs of your organization. In addition to general purpose machines (D series discussed above), Azure offers machines that are optimized for compute (F series), memory (E and M series), or storage (L series). We refer to adding more compute power or memory to a given machine where Tableau Server is deployed as “scaling up” (vertically scaling) your Tableau Server. “Scaling out” (horizontally scaling) refers to adding additional machines to your Tableau deployment, and is discussed in more detail in the High availability strategy section.

Choosing how to scale up—adding more RAM, vCPU, storage—should be determined by the specifics of your organization’s usage patterns. In general, VizQL response times scale with CPU and the Hyper data engine performance scales with RAM. Understanding the pattern of your users (more reliant on extracts, live connections, etc.) will inform the best allocation of resources. Tableau offers several tools designed to help you [test and understand the performance](#) of your Tableau Server. We recommend you use these tools and follow the testing methodology outlined in [this whitepaper](#) to obtain the most accurate results.

High availability strategy

Cloud deployments of business-critical applications like Tableau are designed to take advantage of high availability services and practices. To eliminate single points of failure in solutions, high availability (HA) patterns utilize services to provide redundancy (such as Application Gateway, NAT, or Azure Firewall) or duplicate resources (such as Network Virtual Appliances). There are well established HA patterns in both Azure and Tableau. Each can be tested and tuned for your deployment. Let's start with the Azure HA option.

A public cloud consists of regions which can be geographically grouped (such as US East) or logically grouped (such as Azure Government, broadly speaking). Regions contain a set of data centers, called availability zones (AZ). The classic approach for deploying a cluster in a public cloud is to pick a region and place each node of the cluster in separate data centers. For example, a cluster of three nodes would place a node in AZ1, one in AZ2, and one in AZ3. In this configuration, if the application loses a node—or an AZ because of a data center failure—it is simply the loss of a node to the cluster. This approach is also a [standard HA strategy](#) in other clouds. This approach is [well documented and repeatable](#), offering a 99.99% uptime SLA—but what if there aren't three AZs in the region needed?

In Azure's early growth stage, there weren't always three data centers per region. As an alternative, Microsoft offered an availability solution called Availability Sets. It provides a [replication strategy across compute racks](#) in an AZ as a form of redundancy and offers a 99.95% uptime SLA. Using Availability Sets has historically been encouraged in Azure for this reason. As regions are building out to have a set of three AZs, guidance is moving toward the classic availability zone HA strategy, where nodes are placed in separate AZs.

Now let's turn our attention to Tableau. As Tableau can be deployed as a single node or a cluster, it's time to examine the services Tableau Server deploys and how we might use them to our benefit. Scalability takes two forms—vertical scalability ("scaling up") adds more power to our VMs while horizontal scalability ("scaling out") adds nodes to our cluster. With a VM of reasonable power, general guidance is to scale out first, giving the cluster separate VMs with independence to do certain tasks, like scheduling. In this way, tasks can be isolated to consume resources from one or a subset of nodes, leaving available power to the rest of the cluster for other needs. Deploying Tableau Server across multiple nodes allows you to eliminate single sources of failure, thus achieving a highly available deployment.



Services can be added or removed across a cluster to spread the relevant workloads to optimal compute resources. The good news is that in the cloud, we can test out configurations with our content inexpensively, tune the service mix, and choose a solution we can live with. We highly recommend using this [Tableau Server High Availability whitepaper](#) and our [current Tableau documentation](#) to guide your service and HA planning.

Enterprise DMZ strategy and hub-and-spoke considerations

As on-premises deployments migrated to the cloud, a couple trends emerged. First, the classic **DMZ strategy** (a **subnetwork** that contains and exposes an organization's external-facing services to an untrusted network, generally the Internet) was replicated in VNets in cloud environments, creating confidence in entrusting enterprise deployments there. Secondly, some network devices have become virtualized network virtual appliances (NVAs). A consequence of this has been the proliferation of network rules across disparate VNets and NVAs, network security groups (NSGs) and properties of virtual machines (VMs).

Because of this, the industry has seen the rise of consolidated networking solutions over recent years called next generation firewalls ("next-gen" firewall or NGFW) with hub-and-spoke architectures. **Cisco CSR** has emerged a leading example of this technology, as well as offerings by **Paloalto Networks**, **Fortinet**, and others. Both **Azure Firewall** and **AWS's Transit Gateway** appeared in late 2018 as native NGFW services in their respective public clouds. Each provides a set of consolidated network rules, managed in one place in the "hub" of a hub-and-spoke network topology. Their purpose is to manage and connect multiple VNets and deployments ("spokes") in a simplified, centralized, yet scalable service. Note the use of Azure Firewall in **this DMZ architecture**, which happens to be a hybrid on prem to Azure scenario.

Combining the hub-and-spoke architecture with VNet and network peering allows for flexibility. One Azure Firewall can be set up to manage a number of workloads in Azure. Spokes can be connected to the internet, on-prem, or both. Azure Firewall manages the network rules and Tableau can run as a standalone server or a multi-node deployment in a peered VNet spoke—a process where two VNets connect and exchange traffic. While a **classic DMZ environment** can also be deployed using network virtual appliances (NVAs), we recommend using Azure Firewall due to its secure simplicity and flexibility for your Tableau deployment. Because of these benefits, this whitepaper will describe the Azure Firewall approach.

Microsoft provides a **tutorial for setting up Azure Firewall**. Refer to the Deployment Steps section for details on tailoring these steps in our Tableau HA architecture. Also note the emergence of **Azure Firewall Manager** for globally managed deployments.

Load balancing strategy

Load balancing provides distribution of Tableau work across the nodes of a cluster. It is a good idea to add a load balancer even when deploying a single Tableau Server node. This way, as you expand to more nodes, the load balancer is already set up, IP routing to the cluster remains constant, and the cluster is in the best place to provide a strong SLA to its user base when expanding. The choice of load balancer will not fundamentally affect the DMZ strategy using Azure Firewall.

Now let's discuss our options. We'll start with a quick summary of the networking context and standards at play. Load balancers are generally categorized by their [Open Systems Interconnection](#) (OSI) reference model classification as layer 4 or layer 7. When TCP packets come across the network, a layer 4 process will check the initial information of the packet, but not the packet content. It will perform network address translation (NAT) and forward the packet to the IP address to a backend node. Typically the load balancer will use a round robin method for distributing requests, evenly distributing them across its available nodes. Layer 4 has historically been an inexpensive and effective way of distributing load balancer workloads, but today's CPU and memory resources allow for more sophisticated, yet cost-efficient, options.

Layer 7 is the highest layer (the application layer) in the OSI model, and HTTP belongs in this layer, for example. Layer 7 allows for full packet content inspection and decision-making, and load balancers in this category are empowered to be full reverse proxies. Layer 7 load balancers can make routing decisions on things such as HTTP header content, packet content, and the type of data in the request. This allows for more sophisticated distribution strategies.

Azure offers two options for load balancing. The first is the classic layer 4 [Azure Load Balancer](#), the second is the layer 7 [Application Gateway](#). A third, DNS-based [Traffic Manager](#), will be discussed later. Each is suitable for Tableau Server deployment. Azure Load Balancer is available as basic (free) or standard (not free). For small backend pools (fewer nodes) this may be fine, but note that health probes for SSL are not supported unless you opt for standard. [Azure Load Balancer distribution options](#) can be configured to fit your needs. We provide Application Gateway configuration in the Deployment steps section.

Application Gateway offers the more robust options you'd expect with layer 7. End-to-end SSL, an integrated web application firewall, cookie-based session affinity, and a strong SLA makes it a natural recommendation for external-facing deployments. In published Azure architectures, standard Azure Load Balancer is typically shown for internal use, while Application Gateway is naturally suited for external deployments.

Traffic Manager is a global routing service that offers [broad scope and routing methods](#). Notice its use in [this multi-region reference architecture](#). While Traffic Manager can also be used with a Tableau deployment, this would take the form of Traffic Manager [routing requests to a standard load balancer or Application Gateway](#), which would provide load balancing for a Tableau cluster. So this is an optional service which can be used in broader, global deployments in combination with Application Gateway or the Standard Load Balancer. The Standard Load Balancer and the Application Gateway have similar monthly pricing plus modest data fees. You can explore [pricing options from Azure](#).

Identity and access management

Tableau Server has multiple levels of access security. Authentication verifies a user's identity. Once authenticated, a user is authorized to perform various activities and access certain content within Tableau Server based on their [role](#) and [permissions](#). Finally, row-level security can be enabled to limit a user's access to specific underlying data.

Upon initializing Tableau Server you must select the source of the [identity](#) of your users. This choice can also inform the available options for how to authenticate them. Your selection will depend on the requirements of your organization, the details of your deployment environment and any existing services or protocols you intend to use.

Microsoft offers a set of services for Identity and Access Management (IAM) utilizing Active Directory (AD) and Azure Active Directory (Azure AD). This section will cover the options available for integrating these into your Tableau Server deployment. We focus primarily on Azure AD-based scenarios as these are the most common for customers deploying Tableau Server on Azure. Once you choose a strategy, work with your Azure team and refer to the latest documentation from Azure and Tableau for additional support.

Choosing an identity store

Tableau can be configured for the local identity store or an external identity store using Active Directory (AD) or an arbitrary Lightweight Directory Access Protocol (LDAP) server. In the local identity case, all identities are created and maintained within the Tableau Server repository. For external identity stores, users' identities are copied within the Tableau Server repository in the form of system users, but the external directory is the source of truth about their identity. Both scenarios can be configured to support single sign-on using various protocols.

It is important to distinguish between the Tableau identity store, which is used for defining and provisioning user entities within the Tableau environment, and a generic identity provider, which may be used to actually authenticate a user, through protocols like SAML or OIDC. This leads to two main integration questions for external systems/identity providers:

1. How are we going to provision the users to Tableau?

External identity stores that support the LDAP protocol (such as AD or an arbitrary LDAP directory) support automatic [group sync](#) in Tableau. For all Tableau deployments (regardless of identity store type) you can also always manually add users (individually, or via csv) or sync them via a script utilizing the REST API.

2. How are we going to authenticate them?

The options available to your deployment are influenced by the identity store you choose.

Please note that when configuring Tableau Server, you cannot change the identity store once you have completed installation. If you need to change you will have to uninstall and reinstall Tableau Server. For details, refer to our [authentication documentation](#).

Azure Active Directory and Tableau Server

Azure Active Directory (Azure AD) is Microsoft's cloud-based identity and access management service. Azure AD operates in a SaaS model, meaning all hosting and infrastructure responsibilities are absorbed by Azure. Considering how you might integrate your Tableau Server environment with Azure AD is an important planning task whether you are deploying in a greenfield environment, with an existing Azure AD tenant, or planning to connect your on-premises AD to an Azure AD tenant.

There are two main options for using Azure AD with Tableau Server:

1. Using Azure AD as the external identity store for Tableau Server

1. This is only currently possible if you also utilize [Azure AD Domain Services \(DS\)](#), because Azure AD on its own does not have an LDAP protocol Tableau Server can use to query for users. Domain Services supports LDAP.
2. Azure AD DS is an additional service that you will have to purchase from Microsoft alongside Azure AD to enable this scenario.
3. User provisioning can be done by group-sync from Azure AD.

2. Using Azure AD as the identity provider (idP) for [SAML authentication](#)

1. You can combine this option with number 1, or use this option solely for the authentication piece if you choose not to utilize Domain Services and will provision your users through another means.
2. Some deployments choose to utilize the local identity store and simply write a script to manually sync users via the REST API. Other deployments continue to use their traditional AD environment as the identity store for Tableau Server, but utilize Azure AD for the SAML authentication—see the Active Directory hybrid identity & Tableau Server section.

Active Directory hybrid identity & Tableau Server

Many organizations deploying services to Azure have existing on-premises investments that they are in the process of migrating, or which they would like to maintain. Typically, this on-premises investment includes an Active Directory (AD) tenant which needs to be preserved as a key element of security. For many enterprises, a “lift and shift” to the cloud is a nuanced process that often requires stages of migration and maintaining both on-premises and cloud-based infrastructure. This is commonly referred to as a “[hybrid](#)” scenario. Azure offers additional services and resources that can help you migrate successfully and implement security at every stage:

Azure AD Connect — This service designed to sync user accounts, group memberships, and credential hashes between your local AD tenant and your Azure AD tenant. Using Azure AD connect with Azure AD Domain Services facilitates easier management as updates in Azure AD are synchronized to your on-prem AD. This scenario allows you to leverage both an existing AD tenant and an Azure AD tenant. Refer to Microsoft’s documentation to determine how best to [configure this hybrid scenario for your deployment](#).

Azure Active Directory Domain Services — You can [extend your on-prem AD to Azure](#) or [create a new AD Domain Services forest in Azure](#). These situations apply if you are authenticating for applications both on-prem and in the cloud or if you are in the process of migrating from on-prem to the cloud.

Active Directory Federation Services (ADFS) — This is used to extend enterprise identity store beyond your firewall and provide SSO for off-premises (or cloud-based) services via SAML. You may already be using ADFS to authenticate cloud applications against your on-prem AD tenant. You can host ADFS either via on-prem or on Azure. You can read more about extending Active Directory Federation Services to Azure.

Please refer to Microsoft’s documentation to help you [select the appropriate integration solution](#).

Data source authentication and access

Data source authentication in the context of Tableau Server occurs when a user interacts with a live data source hosted on Tableau Server (by viewing a dashboard, creating a new workbook, connecting to a hosted data source from Tableau Desktop, etc.) or when Tableau Server interacts with a data source on behalf of its users (refreshing an extract).

Data source authentication may be independent from Tableau Server authentication and by default Tableau Server does not act as a proxy to data sources. Just because a user has access to Tableau Server does not mean they are guaranteed access to data sources. You can learn more about [best practices for data sources on Tableau Server](#).

Upon publishing a data source to Tableau Server, you have several options—you can prompt users to enter their own credentials to the data source, embed credentials, or enable an SSO experience that passes user credentials back to the data source. When planning your authentication strategy, especially as it corresponds to data sources, your goal should be to streamline your end user experience (to keep them in their flow) while maintaining appropriate security. You can read detailed explanations about the various options for [determining access to published data sources](#).

When publishing a production data source that multiple users will access, it often makes sense to embed the credentials of a service account from the source database. This account should permit all actions required by Tableau users interacting with the data, but should limit their capabilities for security purposes. When users connect to this Tableau data source they will automatically be authenticated to the back end database under the service account credentials.

Data environment

Data sources

Deploying Tableau Server on Azure allows you to take advantage of existing, or planned investments you already have on Azure in addition to leveraging data solutions outside of the Azure ecosystem.

Tableau [natively connects](#) to Azure SQL Database and Azure SQL Data Warehouse. Together these two services cover the most common data sources for Azure-based organizations. You can configure connections to Azure Blob Storage if your organization is using it as part of a data lake strategy. Tableau also connects natively to on-premises Microsoft-based data sources, which extends to Tableau Server when deployed on Azure. This includes sources like [SQL Server](#), one of our most-used connectors.

Deploying Tableau Server on Azure can also enable your organization's multi-cloud strategy. Tableau offers native connectors to the most popular cloud-based data services—including [Amazon Redshift](#), [Google BigQuery](#), [Databricks](#), and [Snowflake](#). This is in addition to more than 70 native connections Tableau offers. A rise in SaaS-based solutions like Snowflake offer increased deployment flexibility—giving you the ability to choose whether to host on Azure or AWS. Tableau is committed to connecting you to your data, wherever it is. This holds true for deploying Tableau Server on Azure, where you have the flexibility to invest in the data environment that is best for your organization—and with the confidence that Tableau will offer an exceptional user experience no matter your choice.

Data governance and management with Tableau

Data is everywhere and so is the demand to access and analyze it. As our organizations generate more and more data, this process gets increasingly difficult. Effective, integrated data management is the key to bringing order to the chaos. Tableau offers a variety of tools to help accomplish this. [Tableau Prep Conductor](#) allows customers to centralize the scheduling, monitoring, and administration of data preparation performed in [Tableau Prep Builder](#). The new [Tableau Catalog](#) simplifies the discovery of trusted data and content for end users, while also giving IT additional capabilities to manage, monitor, and govern their environment at scale.

In addition to live data connections, Tableau Server also has its own optimized in-memory data engine. [Hyper](#) is designed for fast data ingestion and analytical query processing on large or complex data sets. Leveraging Hyper alongside your live data connections gives you the ability to optimize data performance and decrease analytical loads on your back end systems.

Part of a holistic data strategy is confidence in the security of the data you store within Tableau Server. With 2019.3, we announced support for [encryption at rest](#). This feature supports various compliance scenarios and improves the security of your Tableau Server deployment.

For more in-depth information on data and content governance—including data source and metadata management, curation, content management, and more—read about [governance in Tableau](#).

Planning for operational success

Cost management

Deploying Tableau Server on the cloud offers a variety of operational benefits—flexible scaling, customizable networking, and competitive pricing. As you move to the cloud, it's important to keep these benefits in mind, as well as the increased opportunities for cost optimization. Pricing models are often different from on-premises deployments and careful planning can increase confidence in your migration to the cloud.

Azure offers a variety of sizing options when it comes to the virtual machines where you deploy Tableau Server. Selecting the correct machine size given your deployment is the first step towards managing your cloud costs. As you transition from proof-of-concept or trial deployments to a production deployment it may make sense for your organization to leverage [Azure reserved instances](#). If you know you will be using dedicated virtual machines for a long period of time (one year, three years) you can significantly reduce your costs. Refer back to Virtual machine selection and performance guidance in this paper for more information.

It is important to recognize that deploying Tableau Server on the public cloud in a secure, highly available manner requires additional services beyond virtual machines. Services such as elastic IPs, firewalls, and load balancers all have associated costs. As you plan your deployment make sure to account for the cost of an Azure ecosystem as well as the cost of Tableau Server. You can get more information on the [Azure pricing page](#), and we recommend using [Azure's pricing calculator](#) to better forecast your costs.

Azure also provides detailed cost reporting within the Azure console, as well as documentation on [how best to use cost reporting data](#). We recommend taking several additional approaches to ensure comprehensive and scalable cost management. Implementing [tagging practices](#) within your organization will give you increased control over—and visibility into—where your Azure costs are coming from. Tagging allows you to associate costs with specific departments, teams or projects. Additionally, the data from the cost reporting mentioned above can be exported to Azure Blob storage. Using Azure Data Explorer you can [connect to this data from Tableau](#). For Tableau customers, this offers increased flexibility for analyzing your cost data and centralizing your cost reporting across various platforms.

Data transfer costs

Many public cloud providers, Microsoft Azure included, offer data ingress for free, but typically charge for data egress. If you deploy Tableau Server on Azure and connect to Azure-based data sources, you would not be bringing in any external data and therefore would not pay any data transfer fees. However, if you deploy Tableau Server on Azure and connect to external data sources (for example, Amazon Redshift or Google BigQuery) or upload on-premises data then you will be charged. Most Tableau customers do not find data transfer costs to be prohibitive to deploying Tableau Server on the public cloud—to learn more, see [Azure bandwidth pricing details](#).

It should be noted that Azure also charges for data ingress into an availability zone. If you are following our guidance for highly available deployments, then you will be leveraging Azure's availability zones. These charges are slightly less than the general egress charges. To better forecast your data transfer costs you can leverage Tableau's [administrative views](#) and direct connections to the [Tableau Server Repository](#). These resources will give you insight into how much data is being pulled into Tableau Server and help associate costs with that data transfer. Additionally, monitoring your Azure costs as discussed above can give you increased confidence in forecasting the potential data transfer costs of your deployment.

Automation

One of the benefits of the cloud is the ease of automating deployments and maintenance tasks. Tableau maintains scripts that can be used to [automate installation and configuration](#) of Tableau Server. We have also created a [template for Azure](#) leveraging these scripts that allows you to

quickly deploy a single-node version of Tableau Server on Azure. This serves as both a fast way to deploy Tableau Server on Azure, and a reference for building your own custom Azure and Tableau automation templates.

Tableau Server has a [REST API](#) that can be used to automate administrative tasks once your deployment is running. [Tableau Services Manager](#) provides additional commands to support programmatic administration of your Tableau Server. Together, these tools allow you to automate such tasks as taking backups, programmatically scaling your Tableau Server, and performing upgrades. They can also help you automatically manage users and content (e.g. batch adding users, deleting stale content).

ExpressRoute

As we've discussed, many organizations deploying Tableau Server on Azure have existing on-premises investments they plan to maintain as part of their infrastructure. For these hybrid scenarios you can leverage Azure [ExpressRoute](#)—a secure service to connect your on-premises network to your Azure virtual network. For example, if you are deploying Tableau Server on Azure but will be connecting to an on-premises SQL Server, you may want to consider using ExpressRoute. ExpressRoute offers high speeds, low latency, and does not route data through the public internet. Read [more about ExpressRoute](#) to decide if this service is relevant to your business needs.

Deployment steps

This section documents a Windows deployment. A Linux deployment would share the same HA, load balancer, and Azure Firewall pattern. *Disclaimer: It is intended as a working end to end pattern that can be modified to your organization's specific requirements and standards.*

READ THIS FIRST:

- Create or select the Resource Group you want to use for your deployment. If you can't create one or do not have access to one, work with your Azure team to gain appropriate access to deploy Azure resources. Once created, the Firewall will be in one Resource Group and the Tableau deployment will be in another. Once you have the pattern built, you can adjust details to your requirements and preferences.
- If you don't need the hub-and-spoke architecture, or need a rapid deployment, you'll only build the **tableau-spoke-vnet**. Ignore the Azure Firewall discussion and steps.
- If you want to build the hub-and-spoke architecture, the first step is to build the Azure Firewall. We'll use an exercise provided by Microsoft for [deploying and configuring the Azure Firewall](#), then use it as the hub and peer the Tableau Server spoke VNet with it. Once the Firewall exercise is completed, proceed with the following steps to create the Tableau cluster. Note that this exercise includes a DNS rule that you'll likely omit or configure to your organization's standards.

Create the VNet, subnets, and security group

1. From the Azure portal home page, select Virtual networks. Click Add+ to create a new VNet.
2. Name the VNet **tableau-spoke-vnet**. Unless you need to configure differently, accept the given address space (I'll use 10.6.0.0/16 for this example). Select the appropriate Subscription and Resource Group. Select the Location. Name the initial Subnet **tableau-management-subnet**. Review each item and then click Create.

The screenshot shows the 'Create virtual network' dialog box on the right, overlaid on a list of existing virtual networks on the left. The 'Virtual networks' list includes entries like cb-ha-vnet, Default-Storage-CentralUS, sye-0305-1-vnet, sye-0306-1-vnet, sye-0306-2-vnet, sye-0306-vnet, sye-vnet, tableau-ra-vnet, Test-FW-VN, VN-Hub, and VN-Spoke. The 'Create virtual network' dialog contains the following fields:

- Name: tableau-spoke-vnet
- Address space: 10.6.0.0/16 (10.6.0.0 - 10.6.255.255 (65536 addresses))
- Subscription: cb-azure-wp
- Resource group: cb-azure-wp
- Location: (US) East US
- Subnet:
 - Name: tableau-management-subnet
 - Address range: 10.6.0.0/24 (10.6.0.0 - 10.6.0.255 (256 addresses))
- DDoS protection: Basic (radio button selected)
- Service endpoints: Disabled (radio button selected)
- Firewall: Enabled (radio button selected)

3. Once the VNet is created, go to the VNet. Along its menu, in the Settings section, select Subnets.

Click +Subnet.

The screenshot shows the 'Subnets' blade for the 'tableau-spoke-vnet' virtual network. On the left, a sidebar lists various settings: Overview, Activity log, Access control (IAM), Tags, Diagnose and solve problems, Address space, Connected devices, Subnets (which is selected and highlighted in blue), and DDoS protection. At the top right, there are two buttons: '+ Subnet' and '+ Gateway subnet'. A cursor is hovering over the '+ Subnet' button. Below these buttons is a search bar labeled 'Search subnets'. A table lists one subnet: 'NAME' (tableau-management-subnet) and 'ADDRESS RANGE' (10.6.0.0/24). There is also a 'Search subnets' input field at the bottom of the table.

4. Name it **tableau-workload-subnet**. Accept the default values of the rest of the items and click OK.

We now have our VNet with a management and workload subnet.

The screenshot shows the 'Subnets' blade for the 'tableau-spoke-vnet' virtual network. The left sidebar and top buttons are identical to the previous screenshot. The table now contains two rows of data:

NAME	ADDRESS RANGE	AVAILABLE ADDRESSES
tableau-management-subnet	10.6.0.0/24	251
tableau-workload-subnet	10.6.1.0/24	251

Create the bastion or a jumpbox

There are two options for connecting to your VMs once created. The traditional method is by using a jumpbox, and there is a bastion service available in some regions as well. Here is a walkthrough of each approach.

Bastion connection

The bastion is a [service which provides RDP/SSH connectivity](#) through the Azure portal over SSL. This option eliminates the need for a jump box in the deployment. Unlike using a jumpbox, you do not need to open port 3389 for the bastion service to work.

1. From the Azure portal home page, click +Create a resource. On the next page, type Bastion in the box along the top and select it.
2. Select the subscription and Resource group, and name it **tableau-bastion**, and place it in the appropriate Region and in the **tableau-spoke-vnet**. Note that it will attach to the **AzureBastionSubnet**. Click Review + Create, then Create.

The screenshot shows the 'Create a bastion' wizard in the Azure portal. The steps are:

- Basics**:
 - Subscription: [dropdown]
 - Resource group: cb-azure-wp (with 'Create new' link)
 - Name: tableau-bastion
 - Region: East US
 - Virtual network: tableau-spoke-vnet (with 'Create new' link)
 - Subnet: AzureBastionSubnet (10.6.3.0/24) (with 'Manage subnet configuration' link)
 - Public IP address:
 - Public IP address: Create new (radio button selected)
 - Public IP address name: tableau-spoke-vnet-ip
 - Public IP address SKU: Standard
 - Assignment: Dynamic (radio button)
- Tags**: Not visible in the screenshot.
- Review + create**: The final step before creation.

Jumpbox option

1. From the Azure portal home page, select Virtual machines. Click +Add.
2. Select your subscription and resource group. Name the instance **tableau-jumpbox** and place it in the same Region. No infrastructure redundancy is required. Use a small VM selection, as you simply need this VM for Remote Desktop Protocol (RDP) sessions to your Tableau nodes. We'll use the Windows Server 2016 Datacenter [smalldisk] size for our jumpbox.
3. Enter your Administrator account credentials and Allow selected Port 3389. Click Next : Disks. Accept the defaults and click Next : Networking.

Create a virtual machine

Basics Disks Networking Management Advanced Tags Review + create

Create a virtual machine that runs Linux or Windows. Select an image from Azure marketplace or use your own customized image. Complete the Basics tab then Review + create to provision a virtual machine with default parameters or review each tab for full customization. Looking for classic VMs? [Create VM from Azure Marketplace](#)

PROJECT DETAILS

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

* Subscription

└─ * Resource group cb-azure-wp [Create new](#)

INSTANCE DETAILS

* Virtual machine name tableau-jumpbox ✓

* Region (US) East US

Availability options No infrastructure redundancy required

* Image [smalldisk] Windows Server 2016 Datacenter [Browse all images](#)

* Size Standard DS1 v2
1 vcpu, 3.5 GiB memory [Change size](#)

4. In the Networking menu, place the VM into the **tableau-spoke-vnet** and into the **tableau-management-subnet**. Create a Public IP, give it a name and make it Static. Accept the rest of the settings. At the bottom, click Review + Create.

Create a virtual machine

Basics Disks Networking Management Advanced Tags Review + create

Create a virtual machine that runs Linux or Windows. Select an image from Azure marketplace or use your own customized image.

Complete the Basics tab then Review + create to provision a virtual machine with default parameters or review each tab for full customization.

Looking for classic VMs? [Create VM from Azure Marketplace](#)

PROJECT DETAILS

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

* Subscription: [Subscription dropdown]
└─ * Resource group: cb-azure-wp (dropdown)
 Create new

INSTANCE DETAILS

* Virtual machine name: tableau-jumpbox
* Region: (US) East US
Availability options: No infrastructure redundancy required
* Image: [smalldisk] Windows Server 2016 Datacenter (dropdown)
Browse all images
* Size: Standard DS1 v2
1 vcpu, 3.5 GiB memory
[Change size](#)

5. Go to the **tableau-jumpbox** and note its public and private IP addresses.

Home > cb-azure-wp > tableau-jumpbox

tableau-jumpbox

Virtual machine

Search (Ctrl+F)

Connect Start Restart Stop Capture Delete Refresh

Overview Activity log Access control (IAM) Tags Diagnose and solve problems

Settings Networking

Tags (change) Click here to add tags

Resource group (change): cb-azure-wp	Computer name: tableau-jumpbox
Status: Running	Operating system: Windows (Windows Server 2016 Datacenter)
Location: East US	Size: Standard DS1 v2 (1 vcpu, 3.5 GiB memory)
Subscription (change): [Subscription dropdown]	Public IP address: 13.72.64.253
Subscription ID: e3354229-b7fa-40d4-99aa-ded4fa99ffa1	Private IP address: 10.6.0.5
	Virtual network/subnet: tableau-spoke-vnet/tableau-management-subnet
	DNS name: Configure

Create the first Tableau node

1. From the Azure portal, select Virtual machines, and click +Add. Select the Subscription and Resource group, and name it **tableau-vm1**.
2. Place the VM in the same region. For availability options, select Availability zone, and select 1. For size, select a Windows Server 2016 Datacenter image Standard D16s_v3. Keep Public inbound ports at the default None to prevent access from the internet. Click Next : Disks.

Create a virtual machine

The screenshot shows the 'Create a virtual machine' wizard on the 'Basics' tab. It includes sections for 'PROJECT DETAILS' (Subscription: dbain-at-tableau (EA), Resource group: cb-azure-wp) and 'INSTANCE DETAILS' (Virtual machine name: tableau-vm1, Region: (US) East US, Availability options: Availability zone, Availability zone dropdown showing options 1, 2, 3, selected Standard D2s v3, 2 vcpus, 8 GiB memory). A 'Change size' link is also present.

Basics Disks Networking Management Advanced Tags Review + create

Create a virtual machine that runs Linux or Windows. Select an image from Azure marketplace or use your own customized image.

Complete the Basics tab then Review + create to provision a virtual machine with default parameters or review each tab for full customization.

Looking for classic VMs? [Create VM from Azure Marketplace](#)

PROJECT DETAILS

Select the subscription to manage deployed resources and costs. Use resource groups like folders to organize and manage all your resources.

* Subscription: dbain-at-tableau (EA)

* Resource group: cb-azure-wp

CREATE NEW

INSTANCE DETAILS

* Virtual machine name: tableau-vm1

* Region: (US) East US

Availability options: Availability zone

* Availability zone: 1, 2, 3

* Image: Standard D2s v3
2 vcpus, 8 GiB memory
[Change size](#)

3. Here, we'll create and attach a new disk for the VM. Click on Advanced. Note and keep the Use ephemeral OS disk setting to No. Add a second disk for the Tableau Server. For production installations it is recommended to install Tableau Server on a separate drive of type Premium SSD disk type of at least 128 GB (P10 size). Choose a name and keep Source type as None. Click Ok. Click Next : Networking.

4. In the Networking screen, select the **tableau-spoke-vnet** and the **tableau-workload-subnet**. For the Public IP, select None, and keep the rest of the defaults. We will attach a load balancer to the cluster later, so for now, leave the load balancing solution at No. Click Review + create, and if the review is satisfactory, then Create.

Create a virtual machine

Basics Disks Networking Management Advanced Tags Review + create

Define network connectivity for your virtual machine by configuring network interface card (NIC) settings. You can control ports, inbound and outbound connectivity with security group rules, or place behind an existing load balancing solution. [Learn more](#)

NETWORK INTERFACE

When creating a virtual machine, a network interface will be created for you.

* Virtual network [Create new](#) **tableau-spoke-vnet**

* Subnet [Manage subnet configuration](#) **tableau-workload-subnet (10.6.1.0/24)**

Public IP [Create new](#) **None**

NIC network security group [Create new](#) **None** Basic Advanced

The selected subnet 'tableau-workload-subnet (10.6.1.0/24)' is already associated to a network security group 'tableau-workload-subnet-nsg'. We recommend managing connectivity to this virtual machine via the existing network security group instead of creating a new one here.

Accelerated networking [On](#) On Off

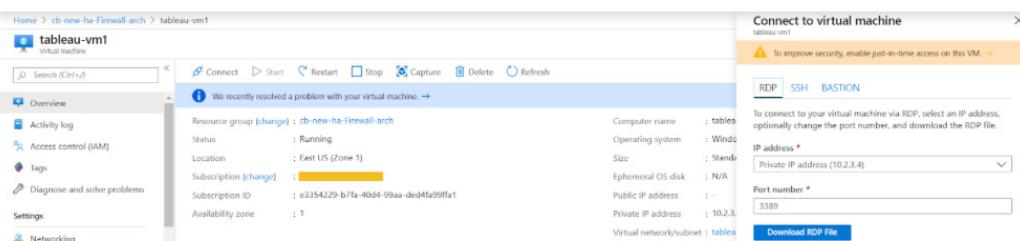
LOAD BALANCING

You can place this virtual machine in the backend pool of an existing Azure load balancing solution. [Learn more](#)

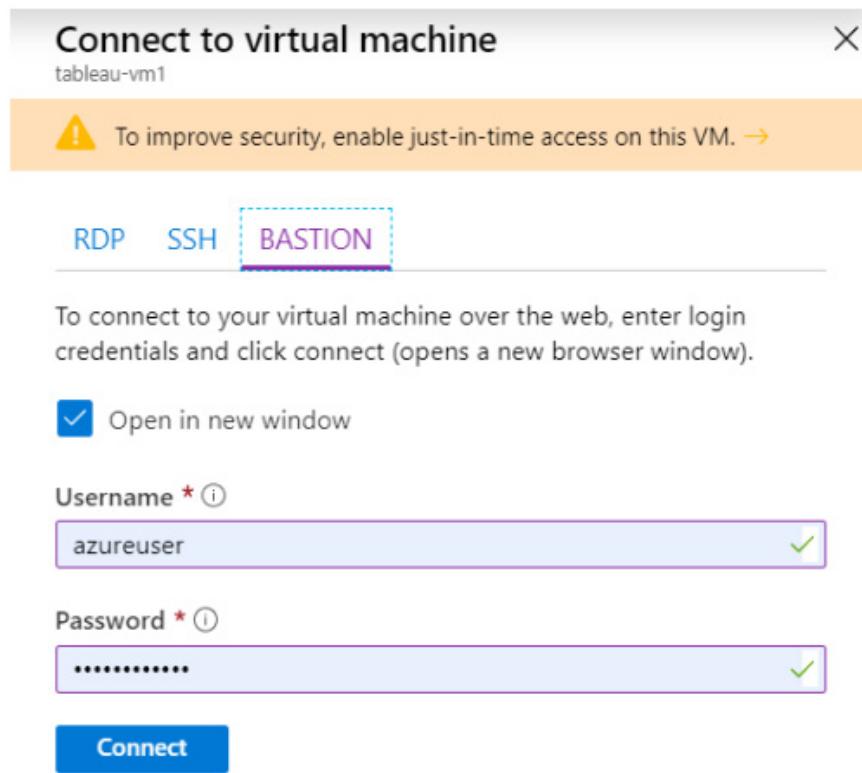
Place this virtual machine behind an existing load balancing solution? Yes No

Connect using bastion

1. Navigate to **tableau-vm1**, and once running, click Connect. Note the Bastion option on the popup menu.



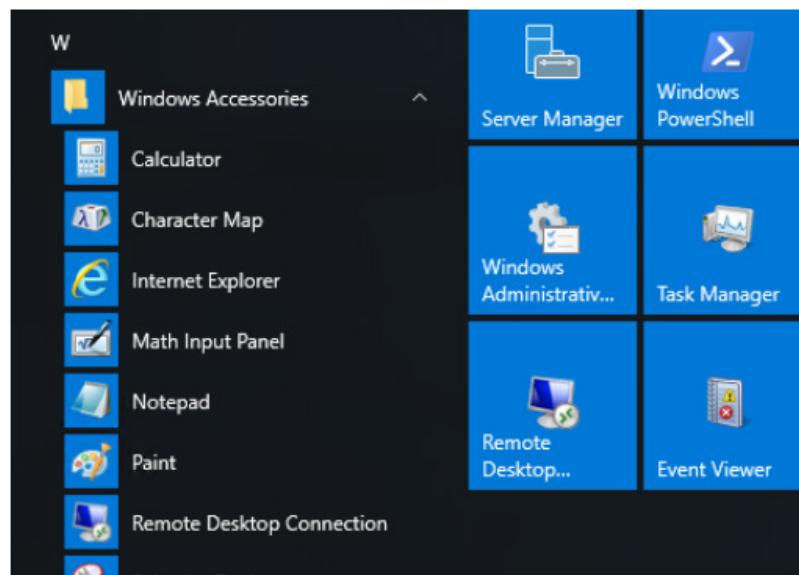
2. Click on Bastion and note the pre-populated Username and Password. Click Connect. You'll see your VM's desktop in your browser.



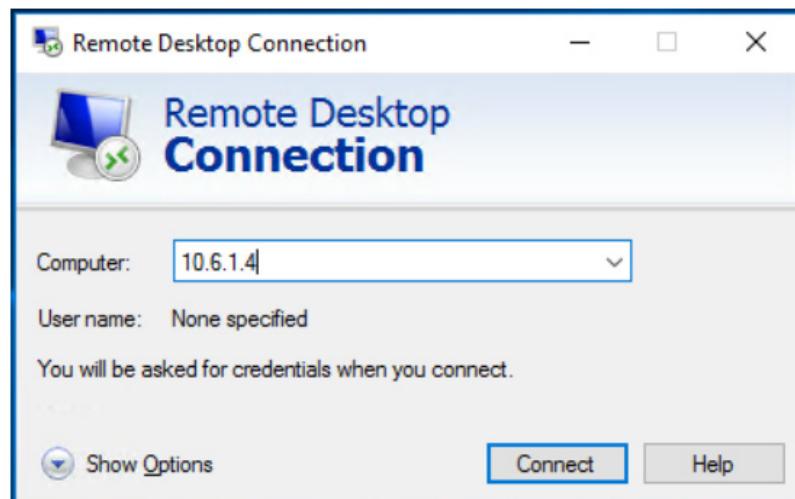
Connect using jumpbox

1. Note the private IP address of the new **tableau-vm1**. You'll use it to access this VM from **tableau-jumpbox**.
2. Return to **tableau-jumpbox**. Click Connect and download the RDP file, which should be called **tableau-jumpbox.rdp**. You will use this to access your Tableau nodes which will have no public IP address. Double click the file and log into the jumpbox using the credentials you provided upon creation. Once logged in, on the blue Networks screen to the right, note the message about allowing the computer to be discoverable on the network, and click No.

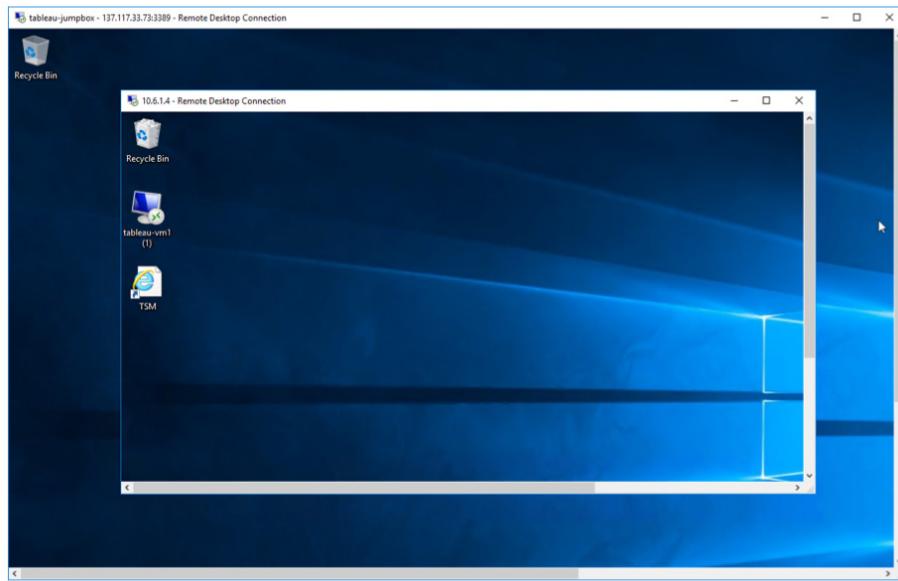
3. Inside the **tableau-jumpbox**, navigate the Windows menu to Windows Accessories, Remote Desktop Connection.



4. Put the private IP address of **tableau-vm1** into the Computer box.

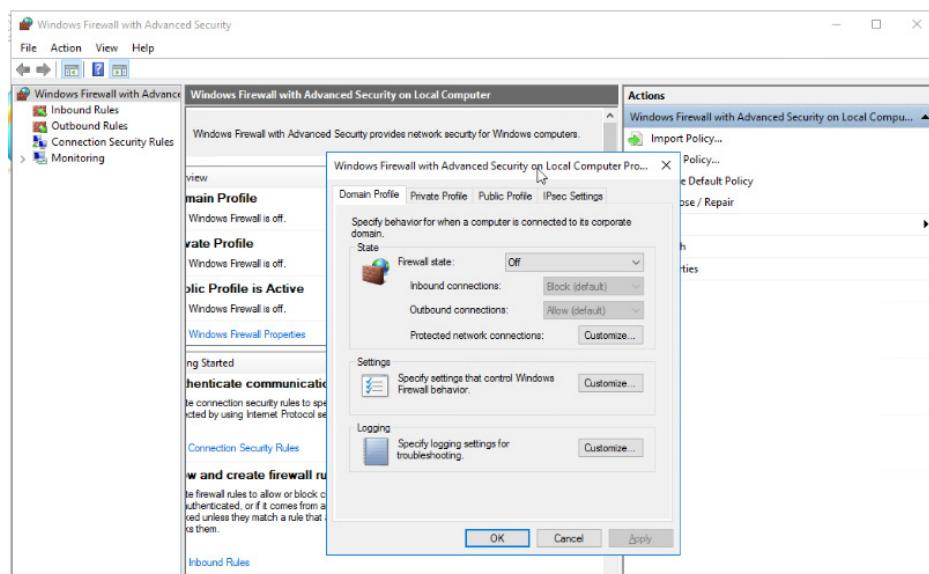


Also, click Show Options, navigate to the Display settings, and move the display settings to less than full screen. (This can be adjusted when in use, but an RDP session within an RDP session in full screen gives the illusion that you are in the jumpbox when you are actually in the Tableau Server node. The adjustment keeps this clearer.)

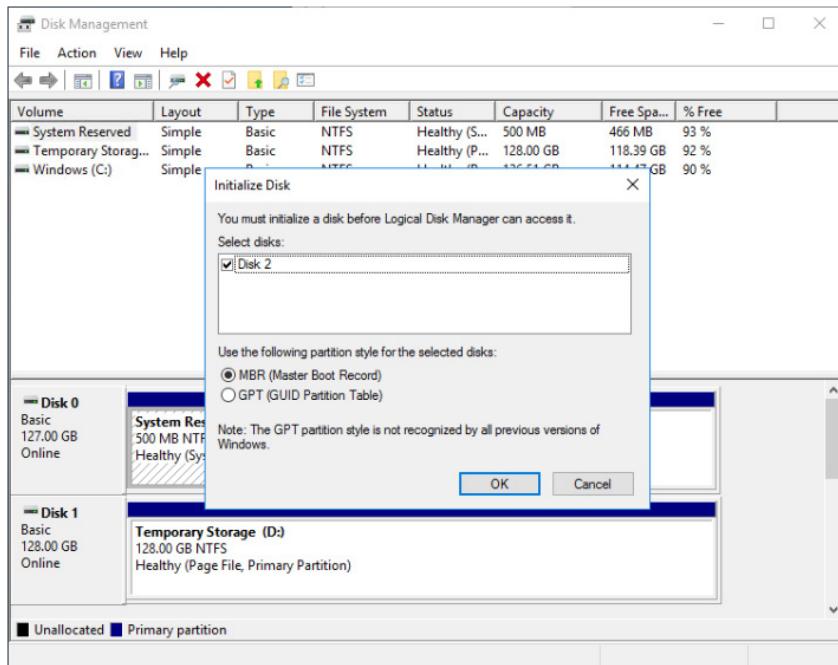


Prepare the VM

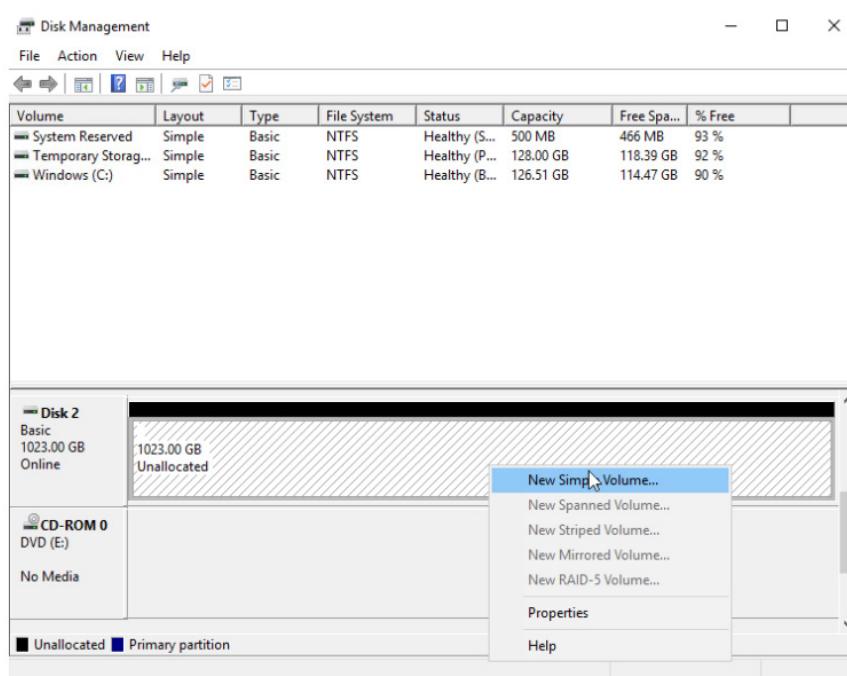
Turn off the Windows Firewall for the Domain, Private, and Public profiles. We'll use the security group and Azure Firewall for that.



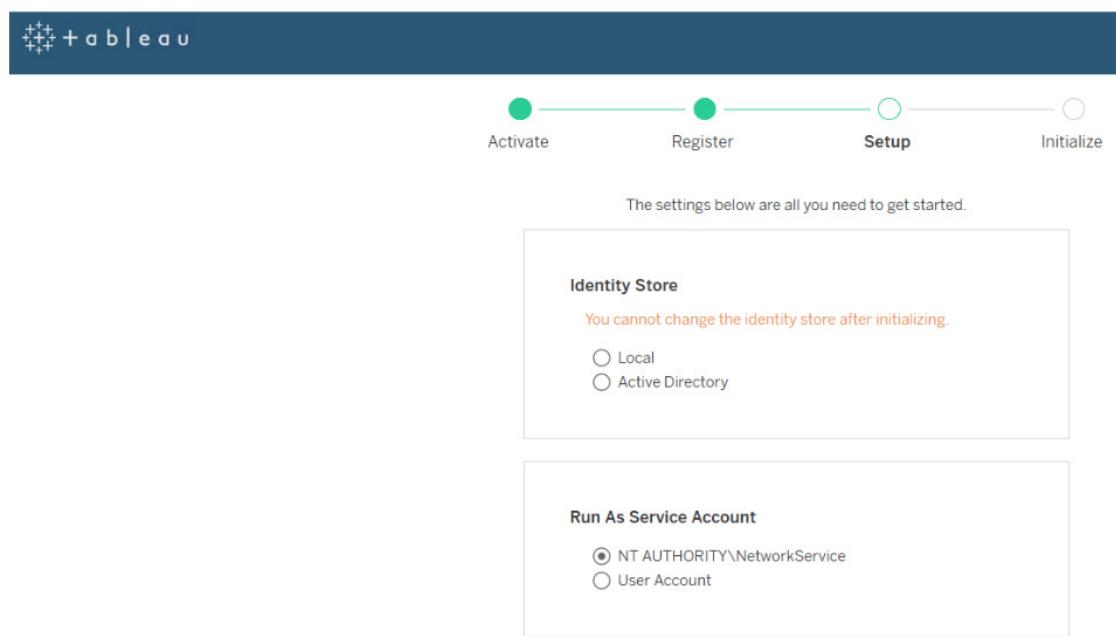
1. In **tableau-vm1**, go to Disk Manager (you can read up on [how to use Disk Manager](#)). Once opened, you'll see that we need to initialize our data disk where we'll install Tableau Server.



Once initialized, hover over the new disk and right click on Disk 2. Select New Simple Volume. Follow the wizard steps and give the new disk a drive letter. I attached the disk to drive F.



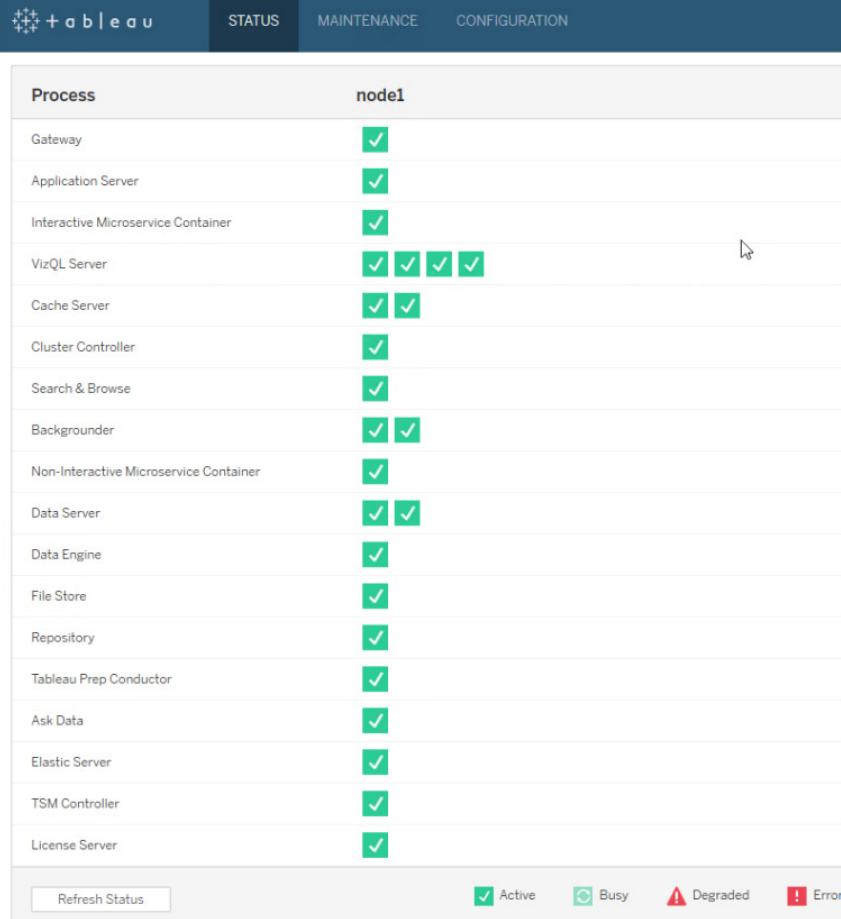
2. Download or copy the Tableau Server installation file to **tableau-vm1**. Install Tableau Server onto your second disk we have attached, following the steps described in the installer. Create a new Tableau Server installation. You can follow the detailed steps in our documentation to [install and configure Tableau](#). Once Tableau Services Manager (TSM) is running, Internet Explorer will open. At this point you can use or generate an SSL certificate for Tableau server if needed. You can find [details on SSL configuration here](#). Log in using the username and password for **tableau-vm1**. After the Register step with your license key, the installation will take you to the Setup tab. Note this is where you can select Active Directory as an identity store—and remember, you will not be able to change this afterwards.



If using the jumpbox, you should download the Tableau installer to the jumpbox, as well as the bootstrap file. These can be copied to the other nodes via RDP in a couple minutes.

3. Installation will proceed for several steps. Once completed, a second browser tab will open and Tableau Server will ask you to specify credentials for creation of the admin account.

4. Validate the Tableau Server is running by viewing the Status menu in the TSM (<https://tableau-vm1:8850/#/status>).



The screenshot shows the Tableau Server Status interface. At the top, there's a navigation bar with icons for Home, Status, Maintenance, and Configuration. The Status tab is selected. Below the navigation bar is a table titled "node1". The table has two columns: "Process" and "Status". The processes listed are: Gateway, Application Server, Interactive Microservice Container, VizQL Server, Cache Server, Cluster Controller, Search & Browse, Backgrounder, Non-Interactive Microservice Container, Data Server, Data Engine, File Store, Repository, Tableau Prep Conductor, Ask Data, Elastic Server, TSM Controller, and License Server. Each process has a status indicator: a green checkmark for Active, a blue circle for Busy, a yellow triangle for Degraded, and a red exclamation mark for Error. Most processes are marked as Active (green checkmark). A cursor is hovering over the VizQL Server row. At the bottom of the table, there are buttons for "Refresh Status" and status legends: Active (green checkmark), Busy (blue circle), Degraded (yellow triangle), and Error (red exclamation mark).

Process	Status
Gateway	✓
Application Server	✓
Interactive Microservice Container	✓
VizQL Server	✓ ✓ ✓ ✓
Cache Server	✓ ✓
Cluster Controller	✓
Search & Browse	✓
Backgrounder	✓ ✓
Non-Interactive Microservice Container	✓
Data Server	✓ ✓
Data Engine	✓
File Store	✓
Repository	✓
Tableau Prep Conductor	✓
Ask Data	✓
Elastic Server	✓
TSM Controller	✓
License Server	✓

Add a NAT gateway

The purpose of the **NAT gateway** is to allow the Tableau node to access the internet, even though it is in a private subnet. This is common practice to allow for software patches, automatic updates, etc. The NAT gateway acts as a proxy out to the internet, and provides this service per availability zone, so if you are deploying a cluster, each zone and VM will need its own NAT gateway. (In the optional hub-and-spoke approach, this functionality can be served by NAT gateways or by Azure Firewall or your firewall of choice for your requirements.)

1. Create a resource, networking, NAT gateway, or search NAT gateway in the Marketplace search.

2. Enter the subscription, resource group, and region information. Select Zone 1 (or the zone of your current node). Name the **NAT tableau-nat1**.

The screenshot shows the 'Create network address translation (NAT) gateway' wizard in the 'Basics' step. The URL is [Home > New > NAT gateway \(Preview\) > Create network address translation \(NAT\) gateway](#). The page title is 'Create network address translation (NAT) gateway'. The 'Subscription' dropdown is set to 'dbain-at-tableau (EA)'. The 'Resource group' dropdown is set to 'cb-nat-test'. Under 'Instance details', the 'NAT gateway name' is 'tableau-nat1', 'Region' is '(US) East US 2', 'Availability zone' is 'Zone 1', and 'Idle timeout (minutes)' is '10'. A note at the bottom right indicates a range from '4-120'.

3. Next, create the IP prefix for the NAT. Be sure to use /31 2 addresses.

The screenshot shows the 'Create network address translation (NAT) gateway' wizard in the 'Outbound IP' step. The URL is [Home > New > NAT gateway \(Preview\) > Create network address translation \(NAT\) gateway](#). The page title is 'Create network address translation (NAT) gateway'. The 'Outbound IP' tab is selected. The 'Public IP addresses' dropdown is set to '(New) nat1' and has a link to 'Create a new public IP address'. The 'Public IP Prefixes' dropdown is set to '0 selected' and has a link to 'Create a new public IP prefix'. An 'Add a public IP prefix' dialog box is open, prompting for 'Name' (set to 'tableaunat1'), 'SKU' (set to 'Standard'), 'IP version' (not specified), 'Prefix size' (set to '/31 (2 addresses)'), and 'Availability zone' (set to '1'). At the bottom of the dialog are 'OK' and 'Cancel' buttons.

4. Specify the **tableau-workload-subnet** which will use the NAT.

The screenshot shows the Azure portal interface for creating a NAT gateway. The top navigation bar includes 'Home', 'New', 'NAT gateway (Preview)', and 'Create network address translation (NAT) gateway'. Below this, a breadcrumb trail shows 'Create network address translation (NAT) gateway'. The main content area has tabs for 'Basics', 'Outbound IP', 'Subnet' (which is selected), 'Tags', and 'Review + create'. A note states: 'Configure which subnets of a virtual network should use this NAT gateway. Subnets with Basic load balancers or virtual machines that are using a Basic public IP are not compatible and cannot be used.' Another note says: 'Note: While you do not have to complete this step to create a NAT gateway, the NAT gateway will not be functional until you have added at least one subnet. You can also add and reconfigure which subnets are included after creating the NAT gateway.' The 'Virtual network' dropdown is set to 'cb-vnet-nat-test'. The 'Subnet' section lists three subnets: 'subnet-nat' (unchecked), 'tableau-workload-subnet' (checked and highlighted), and 'AzureBastionSubnet' (unchecked). The 'Manage subnets >' link is visible. At the bottom, there are buttons for 'Review + create', '< Previous', 'Next : Tags >', and 'Download a template for automation'.

5. Once the NAT is attached to the subnet, it handles outbound communication. To show the NAT working, open a cmd shell in your VM and type ipconfig. Note the IP address(es) of the NAT gateway. Alternately, from the VM, open a browser and go to www.ipchicken.com and check the IP address. It should be one of the two NAT gateway IP addresses.

Create additional nodes as needed

Use the preceding steps and Tableau documentation to install and configure [additional nodes to the cluster](#).

1. Create a second node named **tableau-vm2**. Place it in Availability Zone 2 following the steps from [Create the first Tableau node](#).
2. Following the documentation, log into the TSM on node1. Go to the Configuration tab and download the bootstrap file. Copy and paste it to the jumpbox, along with the original Tableau installation file, then copy them both to **tableau-vm2**. (You can copy them, then paste them onto their RDP sessions. However, using the jumpbox is optional, as it is intentionally a very small VM and space may be tight.)
3. Follow the installation steps to complete the process. Repeat these steps for additional nodes.

Create a load balancer

Now, we'll create a load balancer for the cluster. The Tableau server, and any additional nodes, will be accessed through the load balancer. For convenience, we'll outline both the standard load balancer (OSI layer 4) and the Application Gateway (OSI layer 7) options.

Standard load balancer option

1. From the Azure portal, select Load balancers. Along the top menu, click + Add.
2. Select the Subscription and Resource group as with the other resources, and name it **tableau-internal-lb**. Keep the same Region, and select Internal Type. You'll want to use the Standard SKU. Basic is free, but the key thing to note is that if you need encryption (HTTPS), then you'll want the Standard load balancer, which also provides a static IP address. Select the **tableau-spoke-vnet** and the **tableau-management-subnet**. For Availability Zone, select Zone-redundant for durability. Refer to [load balancer documentation](#) for more information.

The screenshot shows the 'Create load balancer' wizard in the Azure portal. The steps completed are:

- Home > Load balancers > Create load balancer**
- Create load balancer**

PROJECT DETAILS

- * Subscription: [dropdown]
- * Resource group: cb-azure-wp (selected), Create new

INSTANCE DETAILS

- * Name: tableau-internal-lb
- * Region: (US) East US
- * Type: Internal Public
- * SKU: Basic Standard

CONFIGURE VIRTUAL NETWORK.

- * Virtual network: tableau-spoke-vnet
- * Subnet: tableau-management-subnet (10.6.0.0/24)
Manage subnet configuration
- * IP address assignment: Static Dynamic
- * Availability zone: Zone-redundant

At the bottom:

- Review + create**
- < Previous
- Next : Tags >
- Download a template for automation

3. Go to the new **tableau-internal-lb**, note the Settings section and go to the backend pool. Create one and name it **tableau-bep**, and add **tableau-vm1**, **tableau-vm2**, and **tableau-vm3**.

<input checked="" type="checkbox"/> VIRTUAL MACHINE	IP ADDRESS
<input type="checkbox"/> tableau-vm1	ipconfig1
<input type="checkbox"/> tableau-vm2	ipconfig1
<input type="checkbox"/> tableau-vm3	ipconfig1 (10.6.1.6)

Add

Now, from the **tableau-internal-lb** menu, add a health probe. Adjust the interval to 15 seconds, or your organization's architecture standards.

Name
tableau-hp

IP version
IPv4

Protocol
HTTP

Port
80

Path
/

Interval
15 seconds

Unhealthy threshold
2 consecutive failures

OK

5. Finally, add a load balancing rule. Give it a name, and check the HA Ports box. Designate the desired session persistence. We'll select Client IP.

The screenshot shows the 'Add load balancing rule' configuration page for a 'tableau-internal-lb' load balancer. The 'Name' field is set to 'tableau-lbr'. The 'IP Version' is set to 'IPv4'. The 'Frontend IP address' is '10.6.0.6 (LoadBalancerFrontEnd)'. The 'HA Ports' checkbox is checked. The 'Backend pool' is 'tableau-bep (3 virtual machines)'. The 'Health probe' is 'tableau-hp (HTTP:80)'. The 'Session persistence' is set to 'Client IP'. The 'Idle timeout (minutes)' slider is set to 4. The 'Floating IP (direct server return)' option is set to 'Enabled'. At the bottom, there is an 'OK' button.

Finally, follow Azure steps for [integration with the Azure Firewall](#) using asymmetric routing.

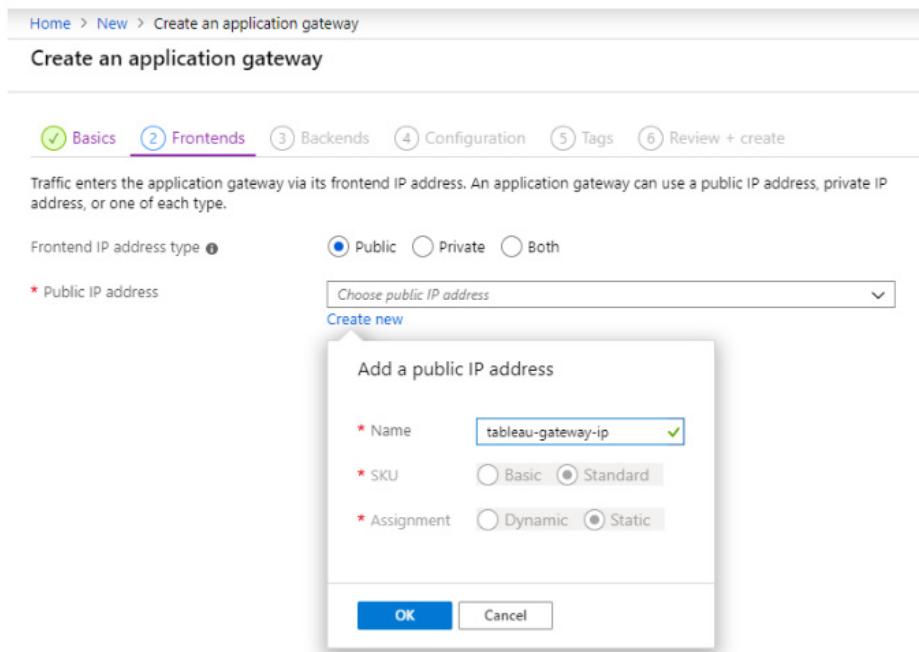
Application Gateway option

Application Gateway provides broad implementation options, like Azure Firewall. This example is provided as a simple, end-to-end working solution with Tableau Server, and modifies [this Quickstart procedure](#). Refer to [Application Gateway documentation](#) and follow your organization's standards for a production configuration.

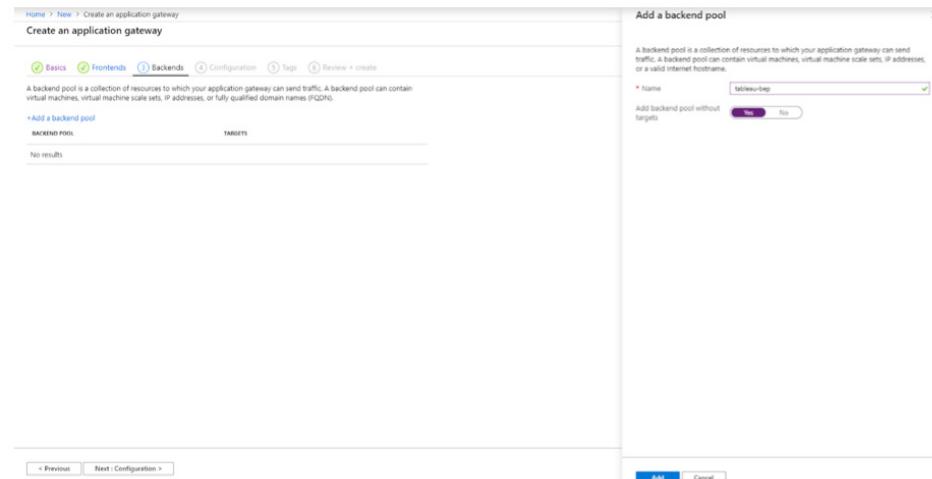
- From the Azure portal, select Create a resource. Select Networking in the New window and then Application Gateway in the list below.
- Select the Subscription and Resource group as with the other resources, and name the new gateway **tableau-agw**. Keep the same Region, and keep the default Standard V2 tier and Enable autoscaling with its 0 to 10 default minimum and maximum instances. Under Virtual network, select **tableau-spoke-vnet**. Click Manage subnet configuration. Click +Subnet to add subnet **tableau-gateway-subnet**. Keep the defaults and click **OK**. Return to Create an application gateway, and select the subnet you created. Click Next : Frontends.

The screenshot shows the Azure portal interface for managing subnets. On the left, the navigation pane includes Home, New, Create an application gateway, tableau-spoke-vnet - Subnets, Overview, Activity log, Access control (IAM), Tags, Diagnose and solve problems, Settings, Address space, Connected devices, Subnets (selected), DDoS protection, Firewall, Security, and Firewall controls. The main area displays the 'Subnets' blade for 'tableau-spoke-vnet'. It lists two existing subnets: 'tableau-management-subnet' (10.6.0.0/24, 250 available addresses) and 'tableau-workload-subnet' (10.6.1.0/24, 249 available addresses). A modal dialog titled 'Add subnet' is open on the right, showing fields for Name ('tableau-gateway-subnet'), Address range (CIDR block) ('10.6.2.0/24'), Network security group ('None'), Route table ('None'), Service endpoints ('None selected'), and Subnet delegation ('Delegate subnet to a service' dropdown set to 'None').

3. In the Frontends screen, select the default Public IP address type. Click Create new and name it **tableau-gateway-ip**. Click OK, then click Next : Backends.



4. On the Backends screen, click **+Add a backend pool**. Name it **tableau-bep**, and click **Yes** to add it without targets for now. Click **Add**, then **Next : Configuration**.



5. On the Configuration page, Add a rule in the Routing rules column. Add a rule called **tableau-routing-rule** and Listener named **tableau-listener**. Select Public for the Frontend IP, with default HTTP protocol. For HTTPS, read more about . For production deployments, implement HTTPS using this [end-to-end SSL encryption with Application Gateway guidance](#). Next, click on the Backend targets tab in the Add a routing rule screen.

6. On the Backend targets tab, create rule **tableau_routing_rule** and select **tableau-beep**.

7. Add an HTTP setting named **tableauHTTP** and keep the defaults. Click Add to return to the Add a routing rule window. Then, click Add to save it and return to the Configuration tab. Click Next : Tags and then Next : Review + create. Review the settings, and if validation passes (green) then click Create.

8. Go to **agw_tableau** and select Backends pools. Then select **tableau-bep**.

NAME	RULES ASSOCIATED
tableau-bep	1

9. Edit the backend pool. Add the Tableau VM nodes from the cluster.

Home > Resource groups > cb-wp-dryrun > agw_tableau - Backend pools > Edit backend pool

Edit backend pool

A backend pool is a collection of resources to which your application gateway can send traffic. A backend pool can contain virtual machines, virtual machine scale sets, IP addresses, or a valid Internet hostname.

★ Name
tableau-bep

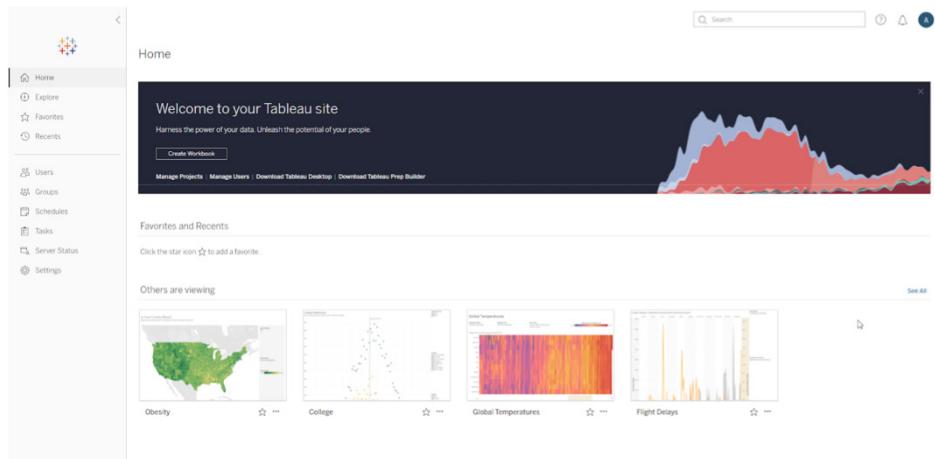
Add backend pool without targets
Yes **No**

Backend targets
2 items

TARGET TYPE	TARGET	⋮
Virtual machine	tableau-vm1732	⋮
Virtual machine	tableau-vm2752 (10.6.1.5)	⋮
IP address or hostname		⋮

Associated rule
[tableau_routing_rule](#)

10. Test the gateway by browsing to its public IP address. Once in, check the Server Status page to see all of the nodes and verify they are working together as configured.



(Optional) Peer the hub and spoke and apply firewall rules

If implementing the hub-and-spoke deployment, now we'll connect the Azure Firewall we created earlier to the Tableau cluster and configure it.

1. Go to the VNet which is the hub containing Azure Firewall (my example is **Test-FW-VN**). Select Peerings from the menu below. Click +Add.

The screenshot shows the 'Test-FW-VN - Peerings' page in the Azure portal. At the top, there's a search bar and a refresh button. Below that, there are two tabs: 'Overview' and 'Activity log'. In the center, there's a table with one row visible, showing a column labeled 'NAME'. At the bottom right of the table area, there's a blue '+ Add' button with a white plus sign, which is highlighted with a dashed blue rectangle and a cursor pointing at it.

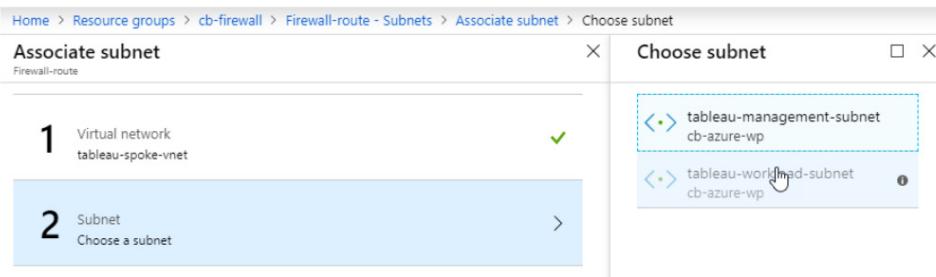
2. Select your subscription and the virtual network you'll peer. Give names to each peering. For clarity we'll use **hub-to-spoke** and **spoke-to-hub**. Click Ok.

The screenshot shows the 'Add peering' configuration page. At the top, it says 'Add peering' and 'Test-FW-VN'. There are several required fields with asterisks:

- * Name of the peering from Test-FW-VN to tableau-spoke-vnet: 'hub-to-spoke' (highlighted with a purple border and a checkmark)
- * Subscription: A dropdown menu showing a yellowed-out option.
- * Virtual network: A dropdown menu showing 'tableau-spoke-vnet (cb-azure-wp)'.
- * Name of the peering from tableau-spoke-vnet to Test-FW-VN: 'spoke-to-hub' (highlighted with a purple border and a checkmark)

Below these, there's a 'Peer details' section with a 'Virtual network deployment model' radio button group (Resource manager selected), a checkbox for 'I know my resource ID', and a 'Configuration' section with a 'Configure virtual network access settings' link and a 'Allow virtual network access from Test-FW-VN to tableau-spoke-vnet' toggle switch set to 'Enabled'.

3. Now, go to the **Firewall-route** in the Resource Group. Select its **Subnets** menu. Click **+Associate** above the right pane. Select **tableau-spoke-vnet** as the VNet and **tableau-workload-subnet** as the subnet. This step provides the route for the peering to the **tableau-spoke-vnet**, and allows the Azure Firewall to manage the rules for it. Click OK.



4. Next, we'll adjust the VM network interfaces to test our Firewall rules with a custom DNS.

Return the Tableau Resource Group. Select **tableau-vm1**. Under Settings, click Networking and note the network interface of each in the upper middle of the page. Click it, and under Settings, select DNS servers. Click Custom and type 209.244.0.3 in the Add DNS server text box, and 209.244.0.4 in the next text box. Repeat these steps for **tableau-vm2** and **tableau-vm3**. Then click Save.

5. Finally, go to the **Test-FW01** Firewall, and under Settings, select Rules. Click Network rule collection along the top menu. Set up the following rule collections:

- Allow-DNS, protocol UDP, which connects your VMs to the DNS server for testing.

```
Source 10.6.1.4, 10.6.1.5, 10.6.1.6
Destination 209.244.0.3, 209.244.0.4
Port 53
```

- Allow-outbound, protocol TCP, which allows traffic from those VMs outbound, including to the internet.

```
Source 10.6.1.4, 10.6.1.5, 10.6.1.6
Destination *
Ports *
```

- If using the Application gateway, create a rule collection inbound_nat. Add the rule gateway-firewall, protocol TCP, which allows traffic to the Firewall.

```
Source *
Destination (Azure Firewall IP address) 52.191.237.104
Ports 80, 443
```

Reminder: These are basic rules to enable a working pattern. You will want to carefully revise and restrict the rules from here to your organization's production requirements. For external deployments always use port 443.

Save and test outbound traffic from each node using the **tableau-jumpbox** as before. You should be able to access the internet.

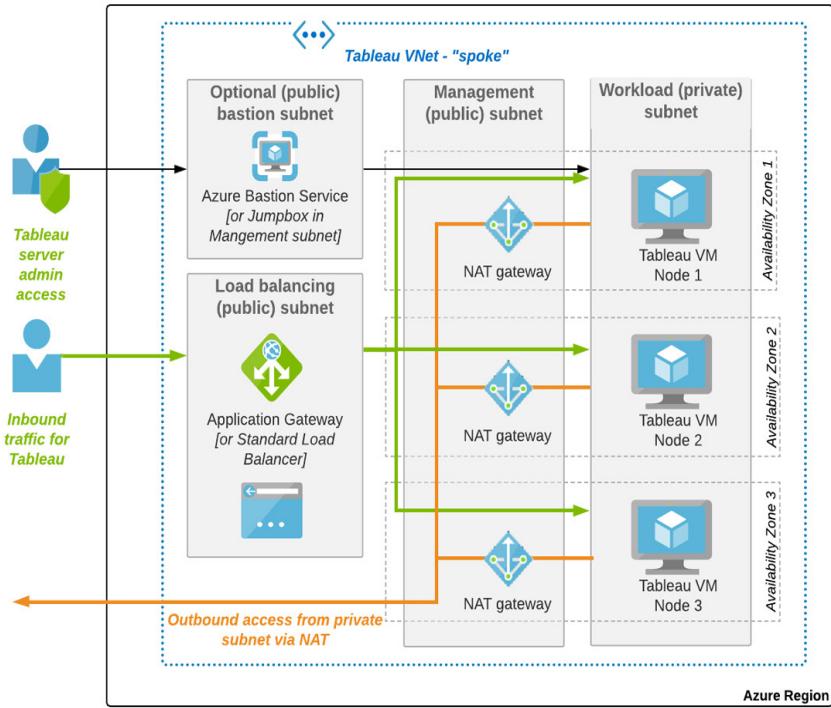
Name	Protocol	Source Addresses	Destination Addresses	Destination Ports
Allow-DNS	UDP	10.6.1.4, 10.6.1.5, 10.6.1.6	209.244.0.3, 209.244.0.4	53
Allow-outbound	TCP	10.6.1.4, 10.6.1.5, 10.6.1.6	*	*

Name	Protocol	Source Addresses	Service Tags	Destination Ports
Allow-outbound	TCP	192.168.10.7, 192.168.10.0/24	0 selected	8080, 8080-8090

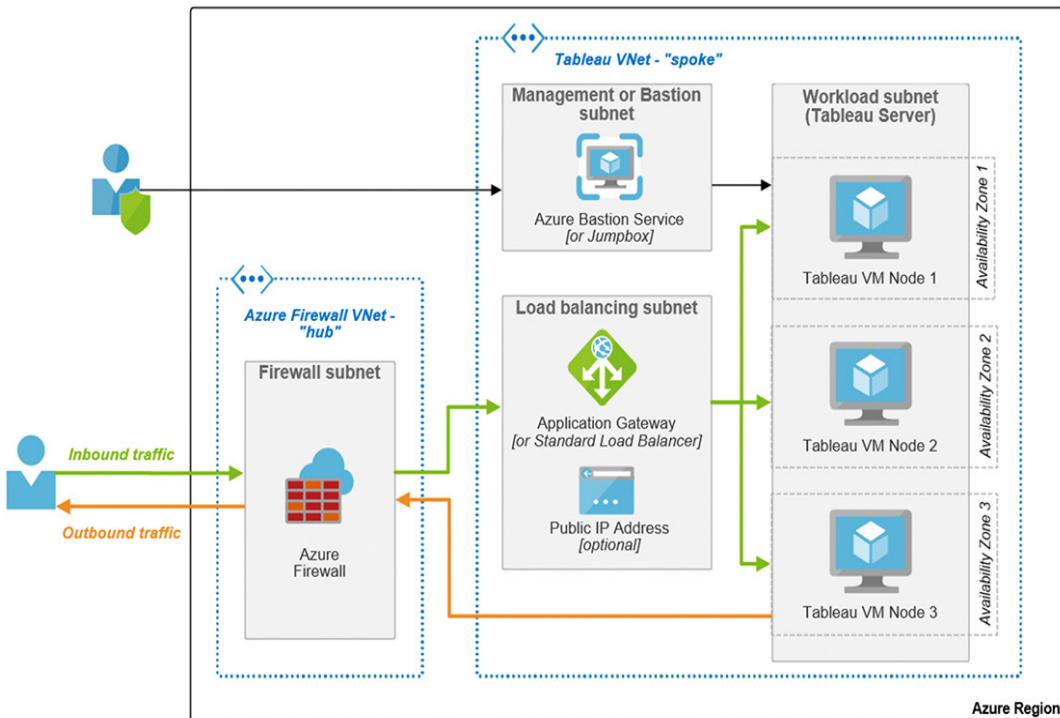
Double-check your routing tables so they support your network configuration—especially between your on-prem and Azure architecture.

Tableau Server on Azure HA reference architecture

1. HA reference architecture



2. Hub-and-spoke version with Azure Firewall



About Tableau

Tableau is a complete, integrated, and enterprise-ready visual analytics platform that helps people and organizations become more data driven. Whether on-premises or in the cloud, on Windows or Linux, Tableau leverages your existing technology investments and scales with you as your data environment shifts and grows. Unleash the power of your most valuable assets: your data and your people.

Additional resources

[Tableau for the Enterprise: IT Overview](#)

[More about Tableau on Microsoft Azure](#)



Acknowledgements

The authors of this paper would like to thank the following individuals for their contributions: Kevin Hulbert, Lauren Kearney, Andrija Marcic, Lee Bond-Kennedy, Robin Cottiss, Conor Knowles, and Spencer Czapiewski.