Horizon Cloud Deployment Guide

For service level starting July 9, 2020
VMware Horizon Cloud Service
You can find the most up-to-date technical documentation on the VMware website at:

https://docs.vmware.com/
Contents

Onboarding to Horizon Cloud for Microsoft Azure, Horizon On-Premises, and Horizon on VMware Cloud on AWS  4

1 Introduction to Horizon Cloud and Onboarding Pods to Become Cloud-Connected Pods  14
   VMware Horizon Cloud Service on Microsoft Azure Requirements Checklist For New Pod Deployments - Updated As Appropriate for Pods That Deploy Starting From the July 2020 Service Release  18
   VMware Horizon Pods with Horizon Cloud - Requirements Checklist - Updated As Appropriate for Connecting Pods Starting From the July 2020 Service Release  34

2 Onboarding Your Very First Cloud-Connected Pod to Your Horizon Cloud Tenant Environment  39
   When Onboarding a Horizon Pod to Use Horizon Subscription Licenses or Cloud-Hosted Services with that Pod  41
   High-Level Workflow When You are Onboarding an Existing Manually Deployed Horizon Pod as Your First Pod to Your Horizon Cloud Tenant Environment  43
   DNS, Ports, and Protocols Requirements When Using Horizon Cloud Connector and a Horizon Pod  48
   Preparing to Run the Onboarding Wizard to Pair a Horizon Pod with Horizon Cloud Using Horizon Cloud Connector  51
   Connect Horizon Cloud Service with an Existing Horizon Pod to Use Horizon Subscription Licenses or Cloud-Hosted Services or Both  56
   When You Choose Microsoft Azure Cloud Capacity for Your Very First Pod Deployment  83
   High-Level Workflow for When Your Very First Horizon Cloud Cloud-Connected Pod is from Using the Pod Deployer to Deploy a Pod into Microsoft Azure  90
   Preparing to Deploy a Horizon Cloud Pod Into Microsoft Azure  96
   Deploy a Horizon Cloud Pod into Microsoft Azure  180
   Troubleshooting If You Encounter Pod Deployment or First-Time Domain Bind Issues  217

3 Now That Your Very First Pod Is Completely Deployed and Connected to Horizon Cloud  240
Onboarding to Horizon Cloud for Microsoft Azure, Horizon On-Premises, and Horizon on VMware Cloud on AWS

These documentation topics here apply when you have just received the Welcome to the VMware Horizon Service email and are ready to onboard your first pod. This onboarding set of topics outlines the relationship between the Horizon universal license and your Horizon Cloud tenant account. The license entitles you to use the cloud-hosted services as well as use the subscription licensing with your pods. This topic and its subtopics also outline the process for your first-time onboarding a pod to Horizon Cloud. This first-time onboarding flow is the key that enables you to exploit your Horizon subscription licenses, onboard your existing Horizon pods to cloud-hosted services, deploy new pods in Microsoft Azure, and leverage all of the cloud-hosted services that VMware Horizon® Cloud Service™ currently provides for cloud-connected pods.

You onboard an existing Horizon pod to the cloud for two primary use cases: to activate a subscription license for that pod and enable your use of those cloud-hosted services that Horizon Cloud provides for that type of pod. You onboard a pod in Microsoft Azure by using the Horizon Cloud Administration Console to deploy that pod into your Microsoft Azure cloud subscription.

**Tip** If you already have at least one cloud-connected pod, instead of this onboarding set of topics, use the companion administration set of topics for information about how to onboard subsequent pods after doing your very first one.

- Relationship Between Onboarding Your First Pod with the Horizon Universal License, the My VMware Account Associated with That License, Your Horizon Cloud Tenant, and the Welcome Email
- Onboarding Requirements Checklists
- Revision History for these Onboarding Topics
- Intended Audience
- About the Screenshots
- Horizon Cloud Community
- Contacting VMware Support
Relationship Between Onboarding Your First Pod with the Horizon Universal License, the My VMware Account Associated with That License, Your Horizon Cloud Tenant, and the Welcome Email

At a high-level, the connect-the-dots between these elements is:

1. Get a subscription license. Currently the Horizon universal license is the license to get. The license will be associated with the specific My VMware account that is used in the license request.

2. VMware sets up the new Horizon Cloud tenant account, associates it with the same My VMware account with which the Horizon universal license is associated, and specifies one of the regional Horizon Cloud control plane instances for the account. The information in the license request is used to determine which regional control plane instance is appropriate for the tenant account. These regional control plane instances are related to the data centers that host the cloud control plane, as described in the service description document available from the Horizon Cloud Service Description and Service Level Agreement page.

3. VMware sends the Welcome to the VMware Horizon Service email to the email address that is configured in the My VMware account in step 1 and with which the license is associated. For an example of this welcome email, see the following screenshot. Among other information, the email states the My VMware account and region that are associated with the tenant account. The stated region appears as one of the following strings: USA, Europe, Australia, USA-2, Europe-2, and Australia-2.

**Note** On June 9, 2020, the Welcome email was updated to standardized regional names. If you received your email before that date, your email contains one of these system-generated strings: USA, EU_CENTRAL_1, AP_SOUTHEAST_2, PROD1_NORTEUROPE_CP1, PROD1_NORTHEUROPE_CP1, and PROD1_AUSTRALIAEAST_CP1.
4 After the email is received, you read the information it contains and use the hypertext links in the Getting Started section to get to key destinations. Those URLs link you to the tenant environment portal (named the Horizon Cloud Administration Console or console for short), the Horizon Cloud Connector software download location, and online documentation.

**Important** After the email is first received, it is prudent to log in to the tenant environment portal using the initial My VMware account and add additional My VMware accounts for those people you want to enable to onboard pods and manage onboarded pods. Adding those people even before onboarding your first pod prevents delays involving timely access to your tenant account. As an example, a delay might occur if the original person is no longer available in your company and you have no one on your team who knows the credentials to log in. Access to your tenant account is needed to onboard pods, as well as to perform related workflows, such as reconfiguring the Horizon Cloud Connector. If access to your tenant account is interrupted as a result of the main person leaving your organization, you would have to open a support request to VMware to update the tenant account's associated My VMware account, which could cause a delay in your logging in to the onboarding and management portals.

For the steps to add additional My VMware accounts to log in to your tenant account, see Add Administrators to Log in to Your Horizon Cloud Tenant Environment.

**Getting the license**

You must get the license first, because that is the point at which VMware generates your Horizon Cloud tenant account and environment.

**Your new Horizon Cloud tenant account in the cloud control plane**
Your Horizon Cloud tenant account is important to you even when your only use case is to use a subscription license with your existing Horizon pods, and not imagining using cloud-hosted services with your pods. The reason why this tenant account is important to you is that same tenant account is use for both:

- Logging in to the Horizon Cloud Connector onboarding and management portal. The Horizon Cloud Connector portal is used for onboarding a Horizon pod to the cloud to use the subscription license, as well as to enable cloud-hosted services. After you complete the initial onboarding of the Horizon pod, you can log in to the Horizon Cloud Connector portal at any time to manage features of the connector itself.

- Logging in to the cloud-based Horizon Cloud tenant environment portal, named the Horizon Cloud Administration Console or console for short. This administrative console is used to add additional administrators so that they can also use the Horizon Cloud Connector onboarding and configuration portal besides the initial My VMware account that got the license. This console is also used to access the cloud-hosted services, such as the Cloud Monitoring Service's monitoring dashboard and reports and the pod deployment wizard for deploying into Microsoft Azure.

How that tenant account relates to the My VMware account associated with the license

A My VMware account must be used to obtain the Horizon universal license in the first place. As a result, that My VMware account is the initial one registered with the newly created Horizon Cloud tenant account and environment, and is used for the login credentials to the Horizon Cloud tenant account (used by both portals listed above). When the tenant account is created, the Welcome to the VMware Horizon Service email is sent to the specific email address that is configured that My VMware account. The following screenshot is an illustration of the welcome mail. You must ensure that you or someone in your organization is able to get the welcome email from that email account which is associated with the My VMware account which was used to purchase the subscription license, so that you’ll can make use of the links in that email to go to downloads for the Horizon Cloud Connector, open the administrative console, and so on.

The following screenshot illustrates a sample of the welcome email and calls out where the My VMware account is referenced.
Your new Horizon Cloud tenant environment and its portal

As soon as you get the welcome email from VMware, the associated My VMware account can log in to that newly created Horizon Cloud tenant environment, even when you have zero cloud-connected pods. However, at this initial point, the administrative console provides access to a single initial screen and a small subset of cloud-hosted workflow actions within that screen.

The following screenshot shows the console at the point in time when your tenant account is first created. The list that follows describes the key actions you can take in that screen prior to onboarding your first pod.
Tip  Click on the General Setup bar in that screen to see two of the key actions that are listed below.

- Learning how to onboard an on-premises Horizon pod, in the **On-Premises** row.

- Learning how to onboard a Horizon pod into your SDDC capacity in VMware Cloud on AWS, from the **Add Cloud Capacity** row’s **Add** button.

- Deploying a pod into Microsoft Azure, from that same **Add Cloud Capacity** row’s **Add** button.
- Adding administrators to whom you want to give the ability to log in to the Horizon Cloud Connector onboarding and configuration portal and the Horizon Cloud Administration Console (the portal to your tenant environment). The My VMware account that was used for setting up the tenant is prefilled there by default. As a result, you will see that row marked with a green check mark. However, that is only because there is always the one initial My VMware account associated with the tenant account when the tenant environment is created.

**Tip**  To prevent being locked out of both your tenant environment and the Horizon Cloud Connector onboarding and configuration portal due to the initial My VMware account going inactive for some reason — such as that person leaving your company or organization — it is prudent to add additional administrators as soon as you receive the **Welcome to Horizon Service** email, even before you onboard a pod for the first time.

- Enablement of the Cloud Monitoring Service (CMS). The CMS is enabled by default, so you will see that row marked with a green check mark. At this point, you can choose to toggle off that feature even prior to onboarding any pods.

**Tip** Before you can access other actions and workflows in that portal beyond the above four, you must have an onboarded pod, that pod must be up and online and communicating with the cloud management plane, and have an Active Directory domain registered with your tenant environment. The console blocks access to other management actions until the Active Directory domain registration workflow is completed. For information about this workflow, see **Performing Your First Active Directory Domain Registration in the Horizon Cloud Environment**.

### Onboarding Requirements Checklists

If your first pod onboarding is to exploit your Horizon subscription license with an existing Horizon pod before starting the steps described in this onboarding documentation set, first read **VMware Horizon Pods with Horizon Cloud - Requirements Checklist - Updated As Appropriate for Connecting Pods Starting From the July 2020 Service Release**. That topic describes various prerequisite elements needed to successfully connect a Horizon pod to Horizon Cloud. After the Horizon pod is cloud-connected, the Horizon subscription license is pushed from the cloud to the pod and you can start enabling the cloud-hosted services for that pod within the administrative console itself.
If your first pod onboarding is in Microsoft Azure, before starting the steps described in this onboarding documentation set, first read VMware Horizon Cloud Service on Microsoft Azure Requirements Checklist For New Pod Deployments - Updated As Appropriate for Pods That Deploy Starting From the July 2020 Service Release. That topic describes various prerequisite elements needed to have a successful first pod deployment into Microsoft Azure.

Revision History for these Onboarding Topics

This documentation set of topics is updated with each release of the product or when necessary. This table provides the update history.

<table>
<thead>
<tr>
<th>Revision</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>8 Sep 2020</td>
<td>Updates made to the screenshots in this guide to align with the availability of the Manage menu on the Getting Started page. Also an update to Horizon Cloud Connector Known Considerations.</td>
</tr>
<tr>
<td>18 AUG 2020</td>
<td>Updates made to align this guide with the Horizon documentation to adopt the terms base image and golden image, as appropriate for each topic’s context.</td>
</tr>
<tr>
<td>5 AUG 2020</td>
<td>Updates made to align this guide with the availability of the new regional cloud control plane in Japan. The following documentation topics are updated for that regional control plane’s DNS names: DNS Requirements for a Horizon Cloud Pod in Microsoft Azure and DNS, Ports, and Protocols Requirements When Using Horizon Cloud Connector and a Horizon Pod.</td>
</tr>
<tr>
<td>9 JUL 2020</td>
<td>Initial version, when the described features were deployed live into production. This document describes the latest service features. For pods using Microsoft Azure capacity, some of the new features are agnostic to the pod manifest version that is newly made available at the time this service release went live into production, while some of the new features require that new pod manifest version for you to take advantage of those features. See the Release Notes for the service 3.1 release for the pod manifest version that corresponds with this service release and for the list of new features that require that latest pod manifest version. For cloud-connected Horizon pods, the latest features correspond to their components at the most recently available VMware Horizon Connection Server software level, connected to the cloud control plane using Horizon Cloud Connector at version 1.7 and later.</td>
</tr>
</tbody>
</table>

Intended Audience

The information in this document is intended for experienced data center administrators with knowledge in the following areas.

- VMware Horizon and VMware Horizon Connection Server
- VMware Horizon Cloud Connector
- VMware Unified Access Gateway™
- VMware Workspace ONE® Access™
- Virtualization technology
- Networking
VMware Cloud™ on AWS (VMware Cloud)
VMware Horizon on VMware Cloud™ on AWS
Microsoft Azure

About the Screenshots

The screenshots typically:

- Show only that portion of the overall user interface screen that corresponds to the text at
  which point the screenshot appears, and not necessarily the full user interface.
- Have blurred areas where appropriate to maintain data anonymity.
- In the PDF format, screenshot images that are wider than 6 inches are automatically resized.
  As a result, such images might appear blurry in the PDF format. In the parallel HTML pages,
  you can click on such wide screenshot images to see the image at its full-size.

**Note** Some screenshots are taken at a higher resolution than others, and might look grainy
when the PDF is viewed at 100%. However, if you zoom to 200%, those images start to look clear
and readable.

Horizon Cloud Community

Use the following communities to ask questions, explore answers given for questions asked by
other users, and access links to useful information.

- VMware Horizon Cloud Service community at https://communities.vmware.com/community/
  vmtn/horizon-cloud-service
- VMware Horizon Cloud on Microsoft Azure sub-community at https://
  communities.vmware.com/community/vmtn/horizon-cloud-service/horizon-cloud-on-azure, a
  sub-community of the VMware Horizon Cloud Service community.

Contacting VMware Support

Contact VMware Support when you need help with your Horizon Cloud environment.

- You can submit a support request to VMware Support online using your My VMware®
  account or by phone.
- KB 2144012 Customer Support Guidelines provides details for getting support depending on
  the issue encountered.
- In the console, clicking > Support, displays the link to that KB 2144012 also.
Selected Pod-Related Terminology Used in these Deployment Guide Documentation Topics

Throughout the Horizon Cloud documentation topics, these phrases have the indicated meanings as follows.

**Horizon pod**

A pod that is constructed based on VMware Horizon product software, and which includes Horizon Connection Server software components.

**Horizon Cloud pod**

A pod that is constructed by running the Horizon Cloud pod deployment wizard which deploys the pod in Microsoft Azure.

**connection broker**

A connection broker is responsible for connecting end users' clients with a virtual desktop VM or farm VM for the purpose of setting up a connected session between each end-user client and the agent running in that VM. This noun — broker — is used because one general definition of the noun broker in the English language is one who negotiates a transaction. In desktop virtualization software's use cases, the connection broker running in a pod receives the end-user's client request to make a connection with a virtual desktop VM or farm VM. The connection broker then negotiates a connected session between the agent running in one of the VMs and that end-user client. The negotiation takes into consideration what types of pod-provisioned resources the end user is entitled to make connections with.

One of the responsibilities of the Horizon Connection Server in a Horizon pod is to serve as a connection broker. For a Horizon Cloud pod deployed in Microsoft Azure, one of the responsibilities of the pod manager VM is to serve as a connection broker.

VMware Technical Publications Glossary

VMware Technical Publications provides a glossary of terms that might be unfamiliar to you. For definitions of terms as they are used in VMware technical documentation, visit http://www.vmware.com/support/pubs.
Introduction to Horizon Cloud and Onboarding Pods to Become Cloud-Connected Pods

Your overall Horizon Cloud tenant environment consists of the VMware-hosted cloud service and your pods deployed into their corresponding capacity environments and connected to the cloud service. When a pod, consisting of VMware software deployed into a supported capacity environment, is appropriately onboarded, then it is a cloud-connected pod. When at least one pod is completely onboarded into your tenant environment, you can onboard additional pods to make for a fleet of cloud-connected pods. To work with your tenant environment's fleet of cloud-connected pods and the desktop-as-a-service features that the service provides, you log in to and use the tenant environment's cloud-based portal, named the Horizon Cloud Administration Console, or the console for short.

Horizon Cloud

A control plane hosted in the cloud by VMware for the central orchestration and management of virtual desktops and applications.

cloud-connected pod

VMware software deployed into a supported capacity environment and onboarded to the cloud control plane. Supported capacity environments are ones such as Microsoft Azure cloud or VMware Cloud™ on AWS or on-premises infrastructure. Each of these capacity environments provides for a specific pod type:

- Pod in your Microsoft Azure subscription
- On-premises Horizon pod
- Horizon pod in VMware Cloud on AWS

Depending on the type of capacity environment you are using, you can use the cloud-based console for an automated pod deployment and connection to Horizon Cloud. For some of those pod types, even though they cannot be automatically deployed and configured, you can still onboard those pods to Horizon Cloud.

For a high-level overview of the pod-onboarding concept, see Onboarding to Horizon Cloud for Microsoft Azure, Horizon On-Premises, and Horizon on VMware Cloud on AWS.
Control Plane

VMware hosts the control plane in the cloud. This control plane provides services that enable the central orchestration and management of virtual desktops, remote desktop sessions, and remote applications for your users. The control plane enables the management of your pods. The pods are physically located in your provided capacity environments. When you log in to the cloud-based console, you can see all of your cloud-connected pods and perform management activities across them, regardless of where they are physically located.

VMware is responsible for hosting the service and providing feature updates and enhancements for a software-as-a-service experience. Horizon Cloud is a multi-tenant environment, and has several regional control plane instances. Each regional control plane instance corresponds to its hosting geographic data center, as described in the service description document available from the VMware Horizon Service Description and Service Level Agreement page. Your tenant account is associated with a specific regional instance at the time the account is created.

The cloud control plane also hosts the common cloud- and web-based management user interface called the Horizon Cloud Administration Console, or console for short. This console runs in industry-standard browsers. It provides IT administrators with a single location for management and administrative tasks involving user assignments and the virtual desktops, remote desktop sessions, and applications. This console is accessible from anywhere at any time, providing maximum flexibility.

**Important**  The administrative console is dynamic and reflects what is available at the current service level. However, when you have cloud-connected pods that are not yet updated to the latest levels of the pod's software, the console does not display those features that depend on the latest pod software level. Also, in a particular release, Horizon Cloud might include separately licensed features or features that are only available for particular tenant account configurations. The console dynamically reflects the elements related to such features only when your license or tenant account configuration includes use of such features. For examples, see Tour of the Cloud-Based Horizon Universal Console for Administrative Tasks in Horizon Cloud.

When you are expecting to see a feature in the administrative console and do not see it, contact your VMware account representative to verify whether your license and tenant account configuration entitles its usage.

Pod Types You Can Connect to Horizon Cloud

This Horizon Cloud release provides for the following deployment types.

**Note**  To connect a pod to Horizon Cloud or use the administrative console for an automated deployment, your customer account must have the appropriate licensing. For licensing information, contact your VMware account representative.
Table 1-1. Pod Deployment Types

<table>
<thead>
<tr>
<th>Deployment Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VMware Horizon pod located in your on-premises infrastructure</td>
<td>Deploy the Horizon Cloud Connector in your on-premises infrastructure and configure it to connect that pod to Horizon Cloud.</td>
</tr>
<tr>
<td>VMware Horizon pod that you manually installed and configured in your VMware Cloud on AWS SDDC</td>
<td>Deploy the Horizon Cloud Connector in your VMware Cloud on AWS SDDC and configure it to connect that pod to Horizon Cloud.</td>
</tr>
<tr>
<td>Horizon Cloud pod deployed by Horizon Cloud into your Microsoft Azure cloud capacity</td>
<td>Deploy the pod using the administrative console's automated deployment wizard.</td>
</tr>
</tbody>
</table>

**Important**  For production environments, ensure the VM models used for your farms and desktop assignments have a minimum of two (2) CPUs. VMware scale testing has shown that using 2 CPUs or more avoids unexpected end-user connection issues. Even though the system does not prevent you from choosing a VM model with a single CPU, you should use such VM models for tests or proof-of-concepts only.
Important  Before launching the pod deployment wizard and starting to deploy your pod, in addition to the requirements below, you must be aware of the following key points:

- Starting with the July 2020 service release, in brand new environments, new pods are required to deploy with at least one gateway configuration. If your customer account was created prior to the July 2020 release, but you have not yet deployed your first pod, deployment of that first pod will require configuring at least one gateway configuration at the time of pod deployment.

- Successful pod deployment requires that none of the Microsoft Azure Policies that you or your IT team have set in your Microsoft Azure environment block, deny, or restrict creation of the pod's components. Also you must verify that your Microsoft Azure Policies Built-in Policy definitions do not block, deny, or restrict creation of the pod's components. As an example, you and your IT team must verify that none of your Microsoft Azure Policies block, deny, or restrict creation of components on Azure storage account. For information about Azure Policies, see the Azure Policy documentation.

- The pod deployer requires that your Azure storage account allow for the deployer to use the Azure StorageV1 and StorageV2 account types. Ensure that your Microsoft Azure Policies do not restrict or deny the creation of content requiring the Azure StorageV1 and StorageV2 account types.

- As part of the pod and gateway deployment processes, Horizon Cloud creates resource groups (RGs) in your Microsoft Azure subscription that do not have tags on them, including the initial resource group that is created for the temporary jump box that orchestrates those deployment processes. Pod deployment will fail if you try to deploy a pod into a Microsoft Azure subscription that has any type of resource tag requirement at the time of deployment, or at the time of pod updates or adding a gateway configuration to a pod. You must verify that your Microsoft Azure Policies allows creation of the pod's untagged resource groups in the target subscription. For the list of RGs that the deployer creates, see the Administration Guide’s Resource Groups Created For a Pod Deployed In Microsoft Azure topic.

- All cloud-connected pods must have line-of-sight to the same set of Active Directory domains at the time you deploy those pods.

This chapter includes the following topics:

- VMware Horizon Cloud Service on Microsoft Azure Requirements Checklist For New Pod Deployments - Updated As Appropriate for Pods That Deploy Starting From the July 2020 Service Release

- VMware Horizon Pods with Horizon Cloud - Requirements Checklist - Updated As Appropriate for Connecting Pods Starting From the July 2020 Service Release
VMware Horizon Cloud Service on Microsoft Azure
Requirements Checklist For New Pod Deployments -
Updated As Appropriate for Pods That Deploy Starting
From the July 2020 Service Release

This checklist will guide you to prepare your Microsoft Azure subscriptions and networks for the
deployment of a pod from Horizon Cloud into Microsoft Azure. Ensuring that these requirements
are fulfilled as described below will provide both for completing a successful new pod
deployment and successfully completing those key tasks that are required to complete after a
pod is deployed.

The sections in this documentation topic are:

- Horizon Cloud Control Plane Requirements
- Microsoft Azure Subscription Requirements
- Network Requirements
- Ports and Protocols Requirements
- Pod Deployment Workflow
- Active Directory Requirements
- Universal Broker Requirements
- DNS Record Requirements
- Horizon Cloud Golden Images, Desktops, and Farms
- Licensing for the Microsoft Windows Operating Systems
- Reference Architecture
- Resources

This checklist is primarily for Horizon Cloud customer accounts that have never had a pod
deployed from or cloud-connected to their tenant environment prior to the July 9, 2020 service
release date. Such environments might be referred to as clean-slate environments or greenfield
environments. New pod deployments that happen after the cloud plane binaries and new pod
manifest versions are pushed into the cloud plane on the July 2020 quarterly service release
date will deploy using the new pod manifest version. The requirements for a successful pod
deployment are primarily determined by the pod manifest version. The cloud plane binaries that
are in the production cloud plane might also determine requirements for a successful
deployment.
Some of the requirements listed below are the ones needed for pod deployment itself. Some requirements are those needed for the key tasks that are performed after pod deployment to get a productive tenant environment, able to provide pod-provisioned desktops and applications to your end users.

Requirements for pod deployment itself

- Horizon Cloud Control Plane Requirements
- Microsoft Azure Subscription Requirements
- Network Requirements
- Ports and Protocols Requirements
- Pod Deployment Workflow

Requirements for a productive environment after pod deployment

- Active Directory Requirements
- Universal Broker Requirements
- DNS Record Requirements
- Horizon Cloud Golden Images, Desktops, and Farms
Before launching the pod deployment wizard and starting to deploy your pod, in addition to the requirements below, you must be aware of the following key points:

- Starting with the July 2020 service release, in brand new environments, new pods are required to deploy with at least one gateway configuration. If your customer account was created prior to the July 2020 release, but you have not yet deployed your first pod, deployment of that first pod will require configuring at least one gateway configuration at the time of pod deployment.

- Successful pod deployment requires that none of the Microsoft Azure Policies that you or your IT team have set in your Microsoft Azure environment block, deny, or restrict creation of the pod's components. Also you must verify that your Microsoft Azure Policies Built-in Policy definitions do not block, deny, or restrict creation of the pod's components. As an example, you and your IT team must verify that none of your Microsoft Azure Policies block, deny, or restrict creation of components on Azure storage account. For information about Azure Policies, see the Azure Policy documentation.

- The pod deployer requires that your Azure storage account allow for the deployer to use the Azure StorageV1 and StorageV2 account types. Ensure that your Microsoft Azure Policies do not restrict or deny the creation of content requiring the Azure StorageV1 and StorageV2 account types.

- As part of the pod and gateway deployment processes, Horizon Cloud creates resource groups (RGs) in your Microsoft Azure subscription that do not have tags on them, including the initial resource group that is created for the temporary jump box that orchestrates those deployment processes. Pod deployment will fail if you try to deploy a pod into a Microsoft Azure subscription that has any type of resource tag requirement at the time of deployment, or at the time of pod updates or adding a gateway configuration to a pod. You must verify that your Microsoft Azure Policies allows creation of the pod's untagged resource groups in the target subscription. For the list of RGs that the deployer creates, see the Administration Guide's Resource Groups Created For a Pod Deployed In Microsoft Azure topic.

- All cloud-connected pods must have line-of-sight to the same set of Active Directory domains at the time you deploy those pods.

### Horizon Cloud Control Plane Requirements

- **Active My VMware account to log in to the Horizon Cloud control plane.**
Microsoft Azure Subscription Requirements

☐ Valid Microsoft Azure subscription in a supported Microsoft Azure environment (Azure Commercial, Azure China, or Azure Government). If you will be deploying the external Unified Access Gateway in a separate VNet using its own subscription, then you need an additional valid Microsoft Azure subscription in the same Microsoft Azure environment.

**Note** Horizon Cloud supports the majority of the Microsoft Azure regions. For the list of Microsoft Azure regions that are not currently supported, see VMware Knowledge Base article Microsoft Azure Regions with Horizon Cloud Service on Microsoft Azure (77121).

☐ Valid Microsoft Azure administrative privileges in that Microsoft Azure subscription. For additional information, see Get Started with Role-Based Access Control in the Azure portal in the Microsoft documentation.

☐ Minimum Microsoft Azure capacity available for Horizon Cloud infrastructure in addition to the expected desktop and app workload. Note that as long as this capacity is made available, Horizon Cloud will automatically deploy these VMs and no manual installation is required.

- Pod Deployment Engine, also known as the Jump Box (transient) — 1 x Standard_F2
- Pod/Pod Manager with High Availability enabled — 2 x Standard_D4_v3 (if no Standard_D4_v3 in the region, 2 x Standard_D3_v2)
- Pod/Pod Manager without High Availability enabled — 1 x Standard_D4_v3 (if no Standard_D4_v3 in the region, 1 x Standard_D3_v2)
- Microsoft Azure Database for PostgreSQL Service — Generation 5, Memory Optimized, 2 vCores, 10 GB Storage
- External Unified Access Gateway (optional, unless no internal gateway is specified) — 2 x Standard_A4_v2 or 2 x Standard_F8s_v2
- Internal Unified Access Gateway (optional, unless no external gateway is specified) — 2 x Standard_A4_v2 or 2 x Standard_F8s_v2

**Note** Starting with the July 2020 service release, in clean-slate environments, new pods are required to deploy with at least one gateway configuration. If your customer account was created prior to the July 2020 release, but you have not yet deployed your first pod, deployment of that first pod will require configuring at least one gateway configuration at the time of pod deployment. As a result, your minimum Microsoft Azure capacity available must accommodate the VMs for one of the gateway configurations, as described in the preceding list.

- If your subscription's region does not provide for the Standard_F8s_v2 VM sizes, the pod deployer wizard does not display that choice in the selector in the Specify Gateways wizard step.
- After pod deployment is finished, your capacity in Microsoft Azure cloud will have to also accommodate the imported VMs, golden images, virtual desktops, and RDSH farms that you create in that pod. When your account is enabled to use the App Volumes features and you use the console's applications capturing workflow, your capacity will have to also accommodate the VMs in the system-generated desktop assignment used for that capturing process. See the Horizon Cloud Golden Images, Desktops, and Farms section below.

The external Unified Access Gateway can be optionally deployed into its own Microsoft Azure Virtual Network (VNet), either using the same subscription as the pod or using a different subscription. When deploying the external Unified Access Gateway into its own VNet, the following capacity is needed in the subscription into which the external Unified Access Gateway is deployed:

- External Unified Access Gateway Deployment Engine, also known as the Gateway Jump Box (transient) — 1 x Standard_F2
- External Gateway Connector — 1 x Standard_A1_v2
- External Unified Access Gateway — 2 x Standard_A4_v2 or 2 x Standard_F8s_v2.
Service principal and authentication key created for each subscription. For additional details, see *Use a portal to create an Azure Active Directory application and service principal that can access resources*. See also *Create the Required Service Principal Needed by the Horizon Cloud Pod Deployer by Creating an Application Registration*.

Each service principal must be assigned the appropriate role that allows for the actions that Horizon Cloud must perform in the subscription. This role can be the Contributor role or a custom role with the required permitted actions at the subscription level. When you are deploying the external gateway configuration in an existing resource group in a separate subscription, more granular permissions can be set for that subscription’s service principal. For additional details about the required role actions, see *Operations Required by Horizon Cloud in Your Microsoft Azure Subscriptions*.

Required resource providers registered in each Microsoft Azure subscription. See step 8.b in *Create the Required Service Principal Needed by the Horizon Cloud Pod Deployer by Creating an Application Registration*.

Microsoft Azure subscription ID, directory ID, application ID and key identified.

If you will be deploying the external Unified Access Gateway in a separate VNet using its own subscription, and your organization requires use of a resource group controlled by you and not created by the pod deployer, you would use the feature to deploy the external Unified Access Gateway into your own named resource group. Use of that feature requires you to create that resource group in that subscription before you run the pod deployer. You also need to ensure the necessary permissions are in place for the pod deployer to deploy the Unified Access Gateway configuration into that resource group, manage the configuration, and update the Unified Access Gateway software in the standard pod update process. For details about the necessary permissions, see *Operations Required by Horizon Cloud in Your Microsoft Azure Subscriptions*.

### Network Requirements

Microsoft Azure Virtual Network (VNet) created in desired Microsoft Azure region with applicable address space to cover required subnets. For additional details, see *Azure Virtual Network*. When deploying the external Unified Access Gateway into its own VNet separate from the pod’s VNet, create that Unified Access Gateway VNet in the same Microsoft Azure region as the pod’s VNet with applicable address space to cover required subnets, and peer the two VNets.

3 non-overlapping address ranges in CIDR format in the pod’s VNet, reserved for subnets.

- Management subnet — 27 minimum
- VM subnet - Primary (tenant) — 27 minimum with /24 - /22 preferred, based on the number of desktops and RDS servers
- DMZ subnet — /28 minimum when Unified Access Gateway is deployed in the pod’s VNet (optional)

Subnets can either be created manually on the VNet or by Horizon Cloud during deployment. If using manually created subnets, no other resources can be attached.

**Note** Starting in the July 2020 quarterly release, for pods deployed new at this release’s manifest level, after the pod is deployed, you can edit the pod to add additional tenant subnets for use with the VMs in your farms and desktop assignments. The additional tenant subnets can be in the same VNet into which the pod is deployed or in a peered VNet. For details, see the *Administration Guide*.
- When deploying the external Unified Access Gateway into its own VNet separate from the pod’s VNet, 3 non-overlapping address ranges in CIDR format in the Unified Access Gateway’s VNet, reserved for subnets:
  - Management subnet — 27 minimum
  - Back end subnet — 27 minimum with /24 - /22 preferred, based on the number of desktops and RDS servers
  - DMZ (front end) subnet — /28 minimum

Subnets can either be created manually on the VNet or by Horizon Cloud during deployment. If using manually created subnets, no other resources can be attached. For this use case, usually the subnets are created manually. In this use case, the back end subnet’s purpose is similar to the purpose of the **VM subnet (Primary)** described in the preceding table row.

- NTP server or servers available and accessible from the Horizon Cloud pod and Unified Access Gateway instances.

- Configure the VNet (Virtual Network) DNS server, pointing to a valid DNS server that can resolve both internal machine names and external names.

- Outbound Internet access on the VNets you are using for the pod and gateway deployment must resolve and reach specific DNS names using specific ports and protocols. This is required for deployment and ongoing operations. For the list of DNS names and ports, see DNS Requirements for a Horizon Cloud Pod in Microsoft Azure and Ports and Protocols Requirements for a Horizon Cloud Pod at the September 2019 Release’s Manifest or Later.

- Proxy server information if required for outbound internet access on the VNet that is used during deployment and ongoing operations of the Horizon Cloud environment (optional)

- Microsoft Azure VPN/Express Route configured (optional)

- FQDN for external user access, internal user access, or both (Required when deploying a pod with Unified Access Gateway).

  **Note** For pods deployed new starting with the July 2020 release, when a pod has both types of gateways, the external gateway and internal gateway configurations on the pod must specify the same FQDN. Because both gateways will use the same FQDN, after pod deployment, you configure Split DNS (Split Domain Name System) to resolve the gateway address either to the external gateway or internal gateway depending on the origin network of the end-user client’s DNS query.
| | Certificate or certificates for Unified Access Gateway in PEM format matching the FQDN (required when deploying a pod with Unified Access Gateway).

**Note**  
- For pods deployed new starting with the July 2020 release, the external gateway and internal gateway configurations on the pod must specify the same FQDN. Because the certificate has to match the FQDN, when a pod has both types of gateways, the same certificate is used in both gateway configurations.
- If the certificate or certificates that you supply for this purpose use CRLs (Certificate Revocation Lists) or OCSP (Online Certificate Status Protocol) settings that refer to specific DNS names, then you must ensure outbound Internet access on the VNet to those DNS names is resolvable and reachable. During configuration of your supplied certificate in the Unified Access Gateway gateway configuration, the Unified Access Gateway software will reach out to those DNS names to check the certificate's revocation status. If those DNS names are not reachable, pod deployment will fail during its Connecting phase. These names are highly dependent on the CA that you used to obtain the certificates, and therefore are not in VMware's control.

| | Two-factor authentication to an on-premises RADIUS authentication server (optional)  
| | - DNS Addresses for Unified Access Gateway to resolve the name of the authentication server  
| | - Routes for Unified Access Gateway to resolve network routing to the authentication server

## Ports and Protocols Requirements

Specific ports and protocols are required for onboarding pods and ongoing operations of your Horizon Cloud environment. See **Ports and Protocols Requirements for a Horizon Cloud Pod at the September 2019 Release's Manifest or Later**.

## Pod Deployment Workflow

The preceding items are the ones needed before you start the pod deployment wizard. After you ensure you have the preceding items, follow the pod deployment steps through step 4 in **High-Level Workflow for When Your Very First Horizon Cloud Cloud-Connected Pod is from Using the Pod Deployer to Deploy a Pod into Microsoft Azure** to deploy your pod.

Then after the pod is successfully deployed, ensure you have the items described in the following section, so that you can complete the remaining key steps in that high-level workflow.

## Active Directory Requirements

The following items are needed for the Active Directory registration workflow. For understanding of that workflow, see **Performing Your First Active Directory Domain Registration in the Horizon Cloud Environment**.
One of the following supported Active Directory configurations:
- On-premises Active Directory Server connected via VPN/Express Route
- Active Directory Server located in Microsoft Azure
- Microsoft Azure Active Directory Domain Services

Supported Microsoft Windows Active Directory Domain Services (AD DS) domain functional levels:
- Microsoft Windows Server 2008 R2
- Microsoft Windows Server 2012 R2
- Microsoft Windows Server 2016

All cloud-connected pods in the same Horizon Cloud customer account must have line-of-sight to the same set of Active Directory domains at the time you deploy those pods. This requirement applies not only to additional pods that you subsequently deploy into Microsoft Azure after the first pod, but also to any on-premises or in-AWS Horizon pods that are cloud-connected to the same customer account using the Horizon Cloud Connector. You can see the checklist for such Horizon pods at VMware Horizon Pods with Horizon Cloud - Requirements Checklist - Updated As Appropriate for Connecting Pods Starting From the July 2020 Service Release.

Domain bind account
- Active Directory domain bind account (a standard user with read access) that has the following permissions:
  - List Contents
  - Read All Properties
  - Read Permissions
  - Read tokenGroupsGlobalAndUniversal (implied by Read All Properties)

Note
- If you are familiar with the VMware Horizon on-premises offering, the above permissions are the same set that are required for the Horizon on-premises offering's secondary credential accounts, stated in this Horizon on-premises documentation topic.
- Generally speaking, the domain bind accounts should be granted the default out-of-the-box read-access-related permissions typically granted to Authenticated Users in a Microsoft Active Directory deployment. However, if your organization's AD administrators have chosen to lock down read-access-related permissions for regular users, you must request those AD administrators preserve the Authenticated Users standard defaults for the domain bind accounts you will use for Horizon Cloud.

You should also set the account password to Never Expire to ensure continued access to log in to your Horizon Cloud environment.

For additional details and requirements, see Service Accounts That Horizon Cloud Requires for Its Operations.
### Auxiliary domain bind account — cannot use the same account as above
- Active Directory domain bind account (a standard user with read access) that has the following permissions:
  - List Contents
  - Read All Properties
  - Read Permissions
  - Read tokenGroupsGlobalAndUniversal (implied by Read All Properties)

**Note**
- If you are familiar with the VMware Horizon on-premises offering, the above permissions are the same set that are required for the Horizon on-premises offering's secondary credential accounts, stated in [this Horizon on-premises documentation topic](#).
- Generally speaking, the domain bind accounts should be granted the default out-of-the-box read-access-related permissions typically granted to Authenticated Users in a Microsoft Active Directory deployment. However, if your organization's AD administrators have chosen to lock down read-access-related permissions for regular users, you must request those AD administrators preserve the Authenticated Users standard defaults for the domain bind accounts you will use for Horizon Cloud.

You should also set the account password to Never Expire to ensure continued access to log in to your Horizon Cloud environment.

For additional details and requirements, see [Service Accounts That Horizon Cloud Requires for Its Operations](#).

### Domain join account
- Active Directory domain join account which can be used by the system to perform Sysprep operations and join computers to the domain, typically a new account (domain join user account)
- Is a member of the Horizon Cloud Administrators Group
- Set account password to Never Expire
- This account requires the following Active Directory permissions: List Contents, Read All Properties, Read Permissions, Reset Password, Create Computer Objects, Delete Computer Objects.
- This account also requires the Active Directory permission named Write All Properties on all descendant objects of the target Organizational Unit (OU) that you plan to use for farms and VDI desktop assignments.
- For additional details and requirements, see [Service Accounts That Horizon Cloud Requires for Its Operations](#).

**Note** In Microsoft Active Directory, when you create a new OU, the system might automatically set the Prevent Accidental Deletion attribute which applies a Deny to the Delete All Child Objects permission for the newly created OU and all descendant objects. As a result, if you explicitly assigned the Delete Computer Objects permission to the domain join account, in the case of a newly created OU, Active Directory might have applied an override to that explicitly assigned Delete Computer Objects permission. Because clearing the Prevent Accidental Deletion flag might not automatically clear the Deny that Active Directory applied to the Delete All Child Objects permission, in the case of a newly added OU, you might have to verify and manually clear the Deny permission set for Delete All Child Objects in the OU and all child OUs before using the domain join account in the Horizon Cloud console.
Auxiliary domain join account (Optional, cannot use the same account as above)
- Active Directory domain join account which can be used by the system to perform Sysprep operations and join computers to the domain, typically a new account (domain join user account)
- Is a member of the Horizon Cloud Administrators Group
- Set account password to Never Expire
- This account requires the following Active Directory permissions: List Contents, Read All Properties, Read Permissions, Reset Password, Create Computer Objects, Delete Computer Objects.
- This account also requires the Active Directory permission named Write All Properties on the all descendant objects of the target Organizational Unit (OU) that you plan to use for farms and VDI virtual desktops.
- For additional details and requirements, see Service Accounts That Horizon Cloud Requires for Its Operations

Note In Microsoft Active Directory, when you create a new OU, the system might automatically set the Prevent Accidental Deletion attribute which applies a Deny to the Delete All Child Objects permission for the newly created OU and all descendant objects. As a result, if you explicitly assigned the Delete Computer Objects permission to the domain join account, in the case of a newly created OU, Active Directory might have applied an override to that explicitly assigned Delete Computer Objects permission. Because clearing the Prevent Accidental Deletion flag might not automatically clear the Deny that Active Directory applied to the Delete All Child Objects permission, in the case of a newly added OU, you might have to verify and manually clear the Deny permission set for Delete All Child Objects in the OU and all child OUs before using the domain join account in the Horizon Cloud console.

Active Directory groups
- Horizon Cloud Administrators — Active Directory security group for Horizon Cloud administrators. Contains the Horizon Cloud administrative users and domain join account. This group is added to the Super Administrators role in Horizon Cloud.
- Horizon Cloud Users — Active Directory security group for the users which will have access to virtual desktops and RDS session-based desktops and published applications in Horizon Cloud.

Active Directory organizational unit (OU) or units (OUs) for virtual desktops and RDS session-based desktops or published applications or both.

Note In Microsoft Active Directory, when you create a new OU, the system might automatically set the Prevent Accidental Deletion attribute which applies a Deny to the Delete All Child Objects permission for the newly created OU and all descendant objects. As a result, if you explicitly assigned the Delete Computer Objects permission to the domain join account, in the case of a newly created OU, Active Directory might have applied an override to that explicitly assigned Delete Computer Objects permission. Because clearing the Prevent Accidental Deletion flag might not automatically clear the Deny that Active Directory applied to the Delete All Child Objects permission, in the case of a newly added OU, you might have to verify and manually clear the Deny permission set for Delete All Child Objects in the OU and all child OUs before using the domain join account in the Horizon Cloud console.

Universal Broker Requirements
Starting with the July 2020 release, when your Horizon Cloud tenant environment is a brand new account starting from that date and you have just completed deploying your first pod into Microsoft Azure, you can choose to use Universal Broker as the brokering method for your environment. When you choose to configure Universal Broker for your environment, the following items are needed. See Configure Universal Broker.
Outbound Internet access on the VNETs you are using for the pod must resolve and reach specific DNS names using specific ports and protocols. This is required for Universal Broker configuration and ongoing operations. For the list of DNS names and ports, see DNS Requirements for a Horizon Cloud Pod in Microsoft Azure and Ports and Protocols Requirements for a Horizon Cloud Pod at the September 2019 Release's Manifest or Later.

Optional: Configure your pod's gateways for two-factor authentication to a RADIUS authentication server, if you want Universal Broker to use two-factor authentication for the pod.
- DNS Addresses for Unified Access Gateway to resolve the name of the authentication server
- Routes for Unified Access Gateway to resolve network routing to the authentication server

Optional: A custom FQDN that your end users will use to access the Universal Broker service and the certificate based on that FQDN (optional)

DNS Record Requirements

After the pod is deployed into the Microsoft Azure cloud and depending on your business situation and the features you want to leverage, it is important to set up records in your DNS server that map fully qualified domain names (FQDNs) to pod-related IP addresses.

**Note** For pods deployed new starting with the July 2020 release, when a pod has both types of gateways, the external gateway and internal gateway configurations on the pod must specify the same FQDN. Because both gateways will use the same FQDN, after pod deployment, you configure Split DNS (Split Domain Name System) to resolve the gateway address either to the external gateway or internal gateway depending on the origin network of the end-user client's DNS query.

If you configured Universal Broker with a custom FQDN, create a public DNS record that maps your custom FQDN to the VMware-provided brokering FQDN in your Universal Broker configuration. See Configure Universal Broker.

Public DNS record created for external end-user access that matches the FQDN, pointing to the Microsoft Azure external load balancer in the pod's external Unified Access Gateway configuration (required when the deployed pod has that configuration). For additional details, see Configuring a custom domain name for an Azure cloud service.

Internal DNS record created for internal end-user access that matches the FQDN, pointing to the Microsoft Azure internal load balancer in the pod's internal Unified Access Gateway configuration (required when the deployed pod has that configuration).
**Internal DNS record created for VMware Workspace ONE Access connections to the pod that matches the certificate that you will upload to the pod itself, pointing to the pod manager's Microsoft Azure internal load balancer. Required when you want to use VMware Workspace ONE Access with the pod.**

**Note** This internal DNS record and a certificate that matches and is uploaded to the pod itself are also used in the rare, atypical use case when you would tell internal end users to directly connect to the pod, instead of telling them to connect to an internal Unified Access Gateway configuration on the pod. Such direct connections are an usual use case. Typically an internal Unified Access Gateway is used instead.

**Certificate chain (CA certificate, SSL certificate, SSL key file) that matches the above internal DNS record created for VMware Workspace ONE Access connections to the pod. For additional details, see [Upload SSL Certificates to a Horizon Cloud Pod for Direct Connections](#). (Also required if you are using the atypical use case of having internal end users connect directly to the pod. Direct connections are a rare, unusual usage. Typically an internal Unified Access Gateway is used instead.)**

---

**Horizon Cloud Golden Images, Desktops, and Farms**

Your Microsoft Azure subscription must accommodate the following requirements depending on the types of golden images, VDI desktops, and RDS farms you want to provision from the deployed pod.

**Note** When your account is enabled to use the App Volumes features and you use the console’s **Capture** action to add App Volumes applications to your inventory, the system generates a desktop assignment of two desktop VMs to support that capturing workflow. Your capacity will have to also accommodate creating those desktops while you are performing the capturing workflow. You can delete that desktop assignment when you are finished capturing applications for your environment.
Base for the golden image — one of the supported Microsoft Azure VM configurations.
- Standard_D3_v2, Standard_D4_v3
- Standard_D2_v2, Standard_D2_v3
- Standard_NV6

Note When using the automated Import VM from Marketplace wizard to create a base VM, the system automatically uses one of the above VM sizes by default. The system's default choice is based on internal algorithms that include checking what sizes are available in the pod's Microsoft Azure region. When using the wizard to create a GPU-enabled VM, the Standard_NV6 size VM is created by default. When using the wizard to create a server OS-based VM, a VM of either Standard_D2_v2 or Standard_D2_v3 is created. When creating a client OS-based VM or a Windows 10 multi-session VM, a VM of either Standard_D3_v2 or Standard_D4_v3 is created.

Model selection for the desktop VMs in VDI desktop assignments — any of the Microsoft Azure VM configurations available in the Microsoft Azure region, except for those not compatible with Horizon Cloud desktop operations.
For production environments, VMware scale testing recommends using models that have a minimum of 2 CPUs or larger.

Model selection for the RDSH VMs in farms — any of the Microsoft Azure VM configurations available in the Microsoft Azure region, except for those not compatible with Horizon Cloud RDS farm operations. This requirement also applies to a VM running Microsoft Windows 10 Enterprise multi-session when that VM is used with Horizon Cloud. As described in the Microsoft Windows 10 Enterprise multi-session FAQ in the Microsoft Azure Windows Virtual Desktop documentation, Microsoft Windows 10 Enterprise multi-session is a Remote Desktop Session Host (RDSH) type that allows multiple concurrent interactive sessions, which previously only Microsoft Windows Server operating systems could provide. The Horizon Cloud workflows that apply to RDS servers are applicable to Microsoft Windows 10 Enterprise multi-session. For production environments, VMware scale testing recommends using models that have a minimum of 2 CPUs or larger.

Licensing for the Microsoft Windows Operating Systems

The items are related to the Microsoft Windows operating systems in your imported VMs, golden images, RDSH-capable farm VMs, and virtual desktop VMs. For the list of Microsoft Windows operating systems that Horizon Cloud supports, see VMware Knowledge Base article 78170 and VMware Knowledge Base article 70965.

Horizon Cloud does not provide any guest operating system licensing required for use of Microsoft Windows operating systems that you use in the course of using the Horizon Cloud workflows. You, the customer, have the responsibility to have valid and eligible Microsoft licenses that entitle you to create, perform workflows on, and operate the Windows-based desktop VMs and RDSH VMs that you choose to use in your Horizon Cloud tenant environment. The required licensing depends on your intended use.

Tip For information about Microsoft Windows Virtual Desktop licensing for Windows 10 Enterprise multi-session and Windows 7 Enterprise, see the Microsoft Azure documentation topic Windows Virtual Desktop pricing.
| ☐ | Licensing for one or more of the following types: Microsoft Windows 7 Enterprise, Microsoft Windows 10 (client types) |
| ☐ | Licensing for Microsoft Windows 10 Enterprise multi-session |
| ☐ | Licensing for one or more of the following types: Microsoft Windows Server 2012 R2, Microsoft Server 2016, Microsoft Server 2019 |
| ☐ | Microsoft Windows RDS Licensing Servers — for high availability, redundant licensing servers are recommended |
| ☐ | Microsoft RDS User or Device CALs or both |

**Reference Architecture**

Use the architecture diagrams below for reference.
Figure 1-1. Illustration of the Horizon Cloud Pod Architecture for a Pod with High Availability Enabled and Configured with Both External and Internal Gateway Configurations, the External Gateway Deployed into the Same VNet as the Pod, Three NICs on the External Gateway VMs, Two NICs on the Internal Gateway VMs, and a Public IP Enabled for the External Gateway’s Load Balancer.
Figure 1-2. Illustration of the External Gateway’s Architecture Elements When the External Gateway is Deployed into Its Own VNet, Separate from the Pod’s VNet, with Three NICS, and into a Resource Group Created by the Pod Deployer

Resources

See the following resources for additional information.

- VMware Horizon Cloud Service documentation page, for links to the product guides and release notes
- VMware Unified Access Gateway documentation page
- Quick Start Tutorial for VMware Horizon Cloud on Microsoft Azure
- Microsoft Azure Resource Manager overview (5 minutes)
- Create Microsoft Azure Service Principal using the portal (6 minutes)
- Microsoft Azure Virtual Network (VNet) (5 minutes)
- Microsoft Azure Virtual Network peering (6 minutes)
VMware Horizon Pods with Horizon Cloud - Requirements Checklist - Updated As Appropriate for Connecting Pods Starting From the July 2020 Service Release

Complete the following tasks to prepare your Horizon pod's components on-premises or in VMware Cloud on AWS for connecting with Horizon Cloud. Ensure every step is completed as described below to complete a successful deployment.

The sections in this documentation topic are:

- Horizon Cloud Control Plane Requirements
- Active Directory Requirements
- Horizon Pod and Horizon Cloud Connector Requirements
- DNS, Ports, and Protocols Requirements
- Universal Broker Requirements
- Licensing for the Microsoft Windows Operating Systems

This checklist is primarily for Horizon Cloud customer accounts that have never had a pod deployed from or cloud-connected to their tenant environment prior to the July 9, 2020 service release date. Such environments might be referred to as clean-slate environments or greenfield environments.

Some of the requirements listed below are the ones needed for successfully onboarding a Horizon pod to Horizon Cloud. Some requirements are those needed for the key tasks that are performed after onboarding the Horizon pod to get a productive tenant environment, able to provide multi-cloud assignments to your end users.

### Horizon Cloud Control Plane Requirements

- ☐ Active My VMware account to log in to the Horizon Cloud control plane.

- ☐ Valid Horizon Universal License. For more information, see the [Horizon Universal License page](#).

### Active Directory Requirements

- ☐ Supported Microsoft Windows Active Directory Domain Services (AD DS) domain functional levels:
  - Microsoft Windows Server 2008 R2
  - Microsoft Windows Server 2012 R2
  - Microsoft Windows Server 2016

- ☐ All cloud-connected pods in the same Horizon Cloud customer account must have line-of-sight to the same set of Active Directory domains at the time you deploy those pods. This requirement applies not only to additional Horizon pods that you subsequently cloud connect using the Horizon Cloud Connector after the first pod, but also to pods deployed into Microsoft Azure using the same customer account. You can see the checklist for Microsoft Azure pods at [VMware Horizon Cloud Service on Microsoft Azure Requirements Checklist For New Pod Deployments - Updated As Appropriate for Pods That Deploy Starting From the July 2020 Service Release](#).
### Domain bind account

- Active Directory domain bind account (a standard user with read access) that has the following permissions:
  - List Contents
  - Read All Properties
  - Read Permissions
  - Read tokenGroupsGlobalAndUniversal (implied by Read All Properties)

**Note**

- If you are familiar with the VMware Horizon on-premises offering, the above permissions are the same set that are required for the Horizon on-premises offering’s secondary credential accounts, stated in [this Horizon on-premises documentation topic](#).
- Generally speaking, the domain bind accounts should be granted the default out-of-the-box read-access-related permissions typically granted to Authenticated Users in a Microsoft Active Directory deployment. However, if your organization’s AD administrators have chosen to lock down read-access-related permissions for regular users, you must request those AD administrators preserve the Authenticated Users standard defaults for the domain bind accounts you will use for Horizon Cloud.

You should also set the account password to *Never Expire* to ensure continued access to log in to your Horizon Cloud environment.

For additional details and requirements, see [Service Accounts That Horizon Cloud Requires for Its Operations](#).

### Auxiliary domain bind account — cannot use the same account as above

- Active Directory domain bind account (a standard user with read access) that has the following permissions:
  - List Contents
  - Read All Properties
  - Read Permissions
  - Read tokenGroupsGlobalAndUniversal (implied by Read All Properties)

**Note**

- If you are familiar with the VMware Horizon on-premises offering, the above permissions are the same set that are required for the Horizon on-premises offering’s secondary credential accounts, stated in [this Horizon on-premises documentation topic](#).
- Generally speaking, the domain bind accounts should be granted the default out-of-the-box read-access-related permissions typically granted to Authenticated Users in a Microsoft Active Directory deployment. However, if your organization’s AD administrators have chosen to lock down read-access-related permissions for regular users, you must request those AD administrators preserve the Authenticated Users standard defaults for the domain bind accounts you will use for Horizon Cloud.

You should also set the account password to *Never Expire* to ensure continued access to log in to your Horizon Cloud environment.

For additional details and requirements, see [Service Accounts That Horizon Cloud Requires for Its Operations](#).
Domain join account

- Active Directory domain join account which can be used by the system to perform Sysprep operations and join computers to the domain, typically a new account (domain join user account)
- Is a member of the Horizon Cloud Administrators Group
- Set account password to Never Expire
- This account requires the following Active Directory permissions: List Contents, Read All Properties, Read Permissions, Reset Password, Create Computer Objects, Delete Computer Objects.
- This account also requires the Active Directory permission named Write All Properties on the OU descendant objects of the target Organizational Unit (OU) that you plan to use for farms and VDI desktop assignments which you create using the Horizon Cloud Administration Console.
- For additional details and requirements, see Service Accounts That Horizon Cloud Requires for Its Operations

Note: In Microsoft Active Directory, when you create a new OU, the system might automatically set the Prevent Accidental Deletion attribute which applies a Deny to the Delete All Child Objects permission for the newly created OU and all descendant objects. As a result, if you explicitly assigned the Delete Computer Objects permission to the domain join account, in the case of a newly created OU, Active Directory might have applied an override to that explicitly assigned Delete Computer Objects permission. Because clearing the Prevent Accidental Deletion flag might not automatically clear the Deny that Active Directory applied to the Delete All Child Objects permission, in the case of a newly added OU, you might have to verify and manually clear the Deny permission set for Delete All Child Objects in the OU and all child OUs before using the domain join account in the Horizon Cloud console.

Auxiliary domain join account (Optional, cannot use the same account as above)

- Active Directory domain join account which can be used by the system to perform Sysprep operations and join computers to the domain, typically a new account (domain join user account)
- Is a member of the Horizon Cloud Administrators Group
- Set account password to Never Expire
- This account requires the following Active Directory permissions: List Contents, Read All Properties, Read Permissions, Reset Password, Create Computer Objects, Delete Computer Objects.
- This account also requires the Active Directory permission named Write All Properties on the OU descendant objects of the target Organizational Unit (OU) that you plan to use for farms and VDI desktop assignments which you create using the Horizon Cloud Administration Console.
- For additional details and requirements, see Service Accounts That Horizon Cloud Requires for Its Operations

Note: In Microsoft Active Directory, when you create a new OU, the system might automatically set the Prevent Accidental Deletion attribute which applies a Deny to the Delete All Child Objects permission for the newly created OU and all descendant objects. As a result, if you explicitly assigned the Delete Computer Objects permission to the domain join account, in the case of a newly created OU, Active Directory might have applied an override to that explicitly assigned Delete Computer Objects permission. Because clearing the Prevent Accidental Deletion flag might not automatically clear the Deny that Active Directory applied to the Delete All Child Objects permission, in the case of a newly added OU, you might have to verify and manually clear the Deny permission set for Delete All Child Objects in the OU and all child OUs before using the domain join account in the Horizon Cloud console.

Active Directory groups

- Horizon Cloud Administrators — Active Directory security group for Horizon Cloud administrators. Contains the Horizon Cloud administrative users and domain join account. This group is added to the Super Administrators role in Horizon Cloud.
- Horizon Cloud Users — Active Directory security group for the users which will have access to virtual desktops and RDS session-based desktops and published applications in Horizon Cloud.
Horizon Pod and Horizon Cloud Connector Requirements

☐ Horizon pod running a minimum of version 7.10 or later. To obtain use of the latest cloud services and features with the cloud-connected pod, it must be running the most currently available version of the Horizon pod software.

☐ Horizon Cloud Connector virtual appliance, a minimum of version 1.5 or later. To obtain use of the latest cloud services and features with the cloud-connected pod, it must be running the most current version, Horizon Cloud Connector version 1.7.
  - Static IP
  - DNS forward and reverse lookup records

☐ Resource requirements for the Horizon Cloud Connector virtual appliance:
  - For version 1.5: 8 vCPUs, 8 GB memory (RAM), 40 GB hard disk
  - For version 1.6: 8 vCPUs, 8 GB memory (RAM), 40 GB hard disk
  - For version 1.7: 8 vCPUs, 8 GB memory (RAM), 40 GB hard disk

**Important** Along with reserving capacity for the Horizon management components such as the Connection Server VMs, Unified Access Gateway VMs, and other components, you should plan on reserving capacity for the Horizon Cloud Connector component. The Horizon Cloud Connector is an infrastructure component that is deployed into your Horizon pod environment to connect a Horizon pod to Horizon Cloud for the use cases of using Horizon subscription licenses and cloud-hosted services with that pod.

☐ Active Directory user used in the pod-onboarding process, when pairing the Horizon Cloud Connector with the pod's Connection Server. This Active Directory user must have the pod's predefined Administrators role on the root access group, as displayed in the pod's Horizon Console in **Global Administrators View > Role Permissions > Administrators**. In other words, the Active Directory user specified for the pod-onboarding process is a super user for that pod, as described in the Horizon documentation's **Horizon Console Administration Guide** that is applicable for your pod's software version.

DNS, Ports, and Protocols Requirements

☐ Specific ports and protocols are required both for onboarding a Horizon pod to Horizon Cloud and for ongoing operations of the pod, the Horizon Cloud Connector paired with that pod, and your Horizon Cloud tenant environment. See **DNS, Ports, and Protocols Requirements When Using Horizon Cloud Connector and a Horizon Pod**.

Universal Broker Requirements

After you complete onboarding your first pod, you can set up use of Universal Broker as the brokering method for your Horizon Cloud environment. When you choose to configure Universal Broker for your environment, at a high-level, the following items are needed. For additional specifics, **Configure Universal Broker** and **System Requirements for Universal Broker**.

☐ To use Universal Broker with a cloud-connected Horizon pod, the pod must have Unified Access Gateway configured.

☐ Universal Broker has specific DNS, port, and protocol requirements to work with participating Horizon pods deployed on premises or in VMware Cloud on AWS. See **Horizon Pods - Port and Protocol Requirements for Universal Broker**.
Optional: Configure your pod's gateways for two-factor authentication to a RADIUS authentication server, if you want Universal Broker to use two-factor authentication for the pod.
- DNS Addresses for Unified Access Gateway to resolve the name of the authentication server
- Routes for Unified Access Gateway to resolve network routing to the authentication server

Optional: A custom FQDN that your end users will use to access the Universal Broker service and the certificate based on that FQDN (optional)

### Licensing for the Microsoft Windows Operating Systems

Horizon Cloud does not provide any guest operating system licensing required for use of Microsoft Windows operating systems that you use in the course of using the Horizon Cloud workflows. You, the customer, have the responsibility to have valid and eligible Microsoft licenses that entitle you to create, perform workflows on, and operate the Windows-based desktop VMs and RDSH VMs that you choose to use in your Horizon Cloud tenant environment. The required licensing depends on your intended use.

- Licensing for one or more of the following types: Microsoft Windows 7, Microsoft Windows 10
- Licensing for one or more of the following types: Microsoft Windows Server 2012 R2, Microsoft Server 2016, Microsoft Server 2019
- Microsoft Windows RDS Licensing Servers — for high availability, redundant licensing servers are recommended
- Microsoft RDS User or Device CALs or both
Onboarding Your Very First Cloud-Connected Pod to Your Horizon Cloud Tenant Environment

You use the topics in this guide when you are starting your journey with VMware Horizon® Cloud Service™ from the very beginning. At the start of this journey, your Horizon Cloud tenant environment starts fresh, without any cloud-connected pods. The first step is to onboard a pod into that fresh environment. That pod will be your very first cloud-connected pod. The topics that follow this one describe how you get that first cloud-connected pod of any one of the pod types currently available for Horizon Cloud.

For a high-level description of the process of obtaining a Horizon Cloud tenant environment and how it relates to onboarding a pod, see Onboarding to Horizon Cloud for Microsoft Azure, Horizon On-Premises, and Horizon on VMware Cloud on AWS.

The following screenshot is a representative example of what a brand new fresh Horizon Cloud environment looks like when you log in to your account for the first time.
That first-time screen is oriented around the idea of adding capacity. You can think of adding capacity here as equivalent to deploying pods in various capacity environments and connecting those pods to your overall Horizon Cloud environment.

<table>
<thead>
<tr>
<th>What the on-screen text says</th>
<th>How that text relates to your first cloud-connected pod</th>
</tr>
</thead>
<tbody>
<tr>
<td>Learn how to add On-Premises capacity...</td>
<td>Use this choice when you want your first cloud-connected pod to be either an existing Horizon pod that you have on premises or that you manually configured in VMware Cloud on AWS. Clicking the link in the user interface informs you to download the Horizon Cloud Connector. Then you follow the steps in the topic High-Level Workflow When You Are Onboarding an Existing Manually Deployed Horizon Pod as Your First Pod to Your Horizon Cloud Tenant Environment.</td>
</tr>
<tr>
<td>Add Cloud Capacity ... Microsoft Azure Cloud</td>
<td>Use this choice when you want your first cloud-connected pod to be one that is created by the wizard-driven, automatic deployment of a pod into Microsoft Azure. When that process is completed, that pod is your first cloud-connected pod. If you are interested in this type for your first cloud-connected pod, learn how to proceed from the topic When You Choose Microsoft Azure Cloud Capacity for Your Very First Pod Deployment and its subtopics.</td>
</tr>
<tr>
<td>Add Cloud Capacity ... VMware Cloud on AWS</td>
<td>Use this choice when you want your first cloud-connected pod to be a Horizon pod located in VMware Cloud on AWS. When your Horizon Cloud customer account has the standard configuration, clicking Add and then the VMware Cloud on AWS choice informs you what to do.</td>
</tr>
</tbody>
</table>
After your first pod is onboarded to your tenant environment, you then use the companion *Horizon Cloud Administration Guide* to read about how to complete the Active Directory domain registration workflow with that first cloud-connected pod. After the domain registration is completed, you continue using that companion *Horizon Cloud Administration Guide* to learn about all of the workflows you can do with Horizon Cloud, including adding additional pods. In that guide, begin with the *Introduction to Horizon Cloud* topic, the *Getting Started Using Your Horizon Cloud Environment*, and their subtopics.

**Note**

- If you have a manually created Horizon pod in VMware Cloud on AWS that you'd like to connect to Horizon Cloud, you can use the add on-premises capacity workflow to connect that existing pod. If that is the first add-capacity workflow you complete, then that pod would become your first cloud-connected pod. Before starting the steps described in this document, see also the VMware Horizon Pods with Horizon Cloud - Requirements Checklist - Updated As Appropriate for Connecting Pods Starting From the July 2020 Service Release. That checklist gives a nice overview of the various prerequisite elements needed to have success with your first pod deployment into VMware Cloud on AWS.

- If you are adding Microsoft Azure cloud capacity as your first cloud-connected pod, before starting the steps described in this document, see also the VMware Horizon Cloud Service on Microsoft Azure Requirements Checklist For New Pod Deployments - Updated As Appropriate for Pods That Deploy Starting From the July 2020 Service Release. That checklist gives a nice overview of the various prerequisite elements needed to have success with your first pod deployment into Microsoft Azure.

This chapter includes the following topics:

- **When Onboarding a Horizon Pod to Use Horizon Subscription Licenses or Cloud-Hosted Services with that Pod**

- **When You Choose Microsoft Azure Cloud Capacity for Your Very First Pod Deployment**

### When Onboarding a Horizon Pod to Use Horizon Subscription Licenses or Cloud-Hosted Services with that Pod

Where the Horizon Cloud user interface refers to on-premises capacity or VMware Cloud on AWS cloud capacity, that means your VMware Horizon pods that you manually installed and configured in your on-premises infrastructure or in your VMware Cloud on AWS SDDC. You onboard such Horizon pods to your Horizon Cloud tenant environment for two primary use cases: to apply a subscription license with your Horizon pod and to use cloud-hosted services with it. When the subscription license is activated on that pod, that activation also provides you the ability to leveraging those cloud-hosted services. Such services include features and workflows for your cloud-connected pods. Both use cases require use of the VMware Horizon® Cloud Connector™, the component that connects your Horizon deployment with the Horizon Cloud cloud-based management plane. That connection provides both for applying the
subscription license to your Horizon deployment and for enabling cloud-hosted services, such as the Cloud Monitoring Service (CMS).

For a high-level description of the process of first-time onboarding any of the supported pod types to the cloud control plane, see Onboarding to Horizon Cloud for Microsoft Azure, Horizon On-Premises, and Horizon on VMware Cloud on AWS.

**Important**  Your Horizon Cloud environment can consist of pods in Microsoft Azure and Horizon pods on-premises and in VMware Cloud on AWS. As a result, all of those cloud-connected pods must have line of sight to the same set of Active Directory domains. If your environment already has pods in Microsoft Azure and you are connecting your first Horizon pod, you must ensure the Horizon pod will be able to have line of sight to the Active Directory domains that are already registered with your Horizon Cloud environment at the time you connect the Horizon pod. For details about the Active Directory domain registration workflow, see Performing Your First Active Directory Domain Registration in the Horizon Cloud Environment.

### Activating Subscription Licenses and Enabling Cloud-Hosted Services for Horizon Pods Using the Horizon Cloud Connector

The Horizon Cloud Connector is a virtual appliance that bridges a Horizon pod with Horizon Cloud. Horizon Cloud Connector is required for using cloud-hosted services with your Horizon pod including Horizon subscription licenses, health status dashboard, and Horizon Help Desk Tool.

**Horizon subscription licenses**

Horizon subscription licenses are available with the standalone Horizon package and as part of the Workspace ONE Enterprise bundle. The Horizon subscription license provides the same product with more flexible deployment options. Horizon subscription licenses enable
Horizon deployment in the data center, private cloud, and in VMware Cloud on AWS. When you have completed the steps using the Horizon Cloud Connector to onboard your pod to Horizon Cloud, VMware activates your subscription license. Within 48 hours, the License Service applies the license to that cloud-connected pod, and a message is displayed in the Horizon administration’s licensing screen:

You must have an active My VMware account to purchase a Horizon license from https://my.vmware.com. You then receive a Welcome email with the link to download the Horizon Cloud Connector as an OVA file.

When you deploy the Horizon Cloud Connector virtual appliance from vSphere Web Client, you pair the Cloud Connector with the Connection Server of the pod you want to connect to Horizon Cloud to use subscription licenses or cloud-hosted services. As part of the pairing process, the Horizon Cloud Connector virtual appliance connects the Connection Server to Horizon Cloud to manage the Horizon subscription license and other services. With a Horizon subscription license, you do not need to manually enter a Horizon license key for the VMware Horizon product activation. However, you do need to use the license keys to activate supporting components such as vSphere, App Volumes, and others.

Cloud-hosted services for Horizon pods

When you have completed the steps using the Horizon Cloud Connector to connect your pod to Horizon Cloud, in addition to applying a subscription license, you can leverage those cloud-hosted services, features, and workflows that Horizon Cloud provides and which are available to you according to that subscription license, such as the Cloud Monitoring Service (CMS). See Introducing the Cloud Monitoring Service’s Unified Visibility, Health Monitoring, and Help Desk Features Provided in Horizon Cloud.

High-Level Workflow When You are Onboarding an Existing Manually Deployed Horizon Pod as Your First Pod to Your Horizon Cloud Tenant Environment

This list is a high-level of the steps when you are onboarding your very first pod to your Horizon Cloud tenant environment and that pod is an existing manually deployed Horizon pod. A manually deployed pod is one that you manually installed and configured using either on-premises capacity or using VMware Cloud on AWS capacity. After these onboarding steps are completed for your very first cloud-connected pod, the subscription license is applied to that onboarded Horizon pod and you can start using the cloud-hosted services that Horizon Cloud
provides for that pod type, which include the Cloud Monitoring Service (CMS). At that point, you can also onboard additional pods.

The following diagram illustrates the overall flow.

Before beginning this workflow, you must have already installed and configured your Horizon pod. For information about manually installing a Horizon pod that you can use with this Horizon Cloud release:

- For manually installing pods using on-premises capacity, see the relevant installation information for the most recent version of VMware Horizon from the Horizon Documentation page.

- For manually installing pods using VMware Cloud on AWS capacity, see the best practices guide for deploying Horizon pods on VMware Cloud on AWS, available from the Horizon on VMware Cloud on AWS product page.
You onboard an existing Horizon pod to the cloud for two primary use cases: to activate a subscription license for that pod and enable your use of those cloud-hosted services that Horizon Cloud provides for this type of pod, such as the Cloud Monitoring Service (CMS). The CMS is one of the central services provided in Horizon Cloud. CMS provides for visibility, health monitoring, and help desk services with cloud-connected pods. For a high-level description of the process of onboarding a pod to the cloud control plane, see [Onboarding to Horizon Cloud for Microsoft Azure, Horizon On-Premises, and Horizon on VMware Cloud on AWS](#).

**Caution** Complete all of the steps below to fully connect your first pod to Horizon Cloud before you start deploying the Horizon Cloud Connector with any subsequent manually installed pod you want to connect. Due to a known issue in this release, if you finish connecting more than one pod to the cloud using the Horizon Cloud Connector before you complete the Active Directory domain registration and Super Administrators role assignment step at least once, the Active Directory domain registration step will fail. At that point, you will have to unplug all but one of your cloud-connected Horizon pods before you can successfully complete the required Active Directory domain registration and Super Administrators role assignment step.

1. Fulfill the prerequisites, which includes obtaining a Horizon subscription license, such as the Horizon universal license. See [VMware Horizon Pods with Horizon Cloud - Requirements Checklist - Updated As Appropriate for Connecting Pods Starting From the July 2020 Service Release](#) as well as the prerequisites section in [Connect Horizon Cloud Service with an Existing Horizon Pod to Use Horizon Subscription Licenses or Cloud-Hosted Services or Both](#).

2. Verify you meet the DNS, ports, and protocol requirements for connecting a Horizon pod with Horizon Cloud. See [DNS, Ports, and Protocols Requirements When Using Horizon Cloud Connector and a Horizon Pod](#).

3. If your environment requires use of a proxy server for the Horizon Cloud Connector virtual appliance to reach the Internet, obtain the required proxy settings so that you can specify them when you deploy the appliance's OVF template.

**Important** If you plan to use Universal Broker with Horizon Cloud Connector 1.5, and require using proxy settings, you must configure those proxy settings when you deploy the OVF template. Due to a known limitation, the Universal Broker will not pick up changes to proxy settings made in the Horizon Cloud Connector appliance after the initial deployment process. Due to a known issue, the system is not honoring any no-proxy host configuration that you specified in the initial OVF template deployment. To use a no-proxy host configuration with the Horizon Cloud Connector appliance, you must configure it after the initial deployment process. See [Modifying Proxy Settings for Horizon Cloud Connector 1.5 or Earlier](#) for more details.
4 Optionally log in to the Horizon Cloud tenant portal and configure additional administrators for your tenant environment.

   **Tip** Even though you can complete the next steps to onboard the pod solely using the My VMware account that is the one initially associated with your tenant environment, it is prudent to configure additional administrators at the start of this process. If only a single My VMware account is associated with your tenant account and you lose access to the credentials, delays might occur because you'll have to open a service request with VMware to associate a new My VMware account with the tenant account. To prevent such delays, log in to the tenant portal at cloud.horizon.vmware.com with the initially associated My VMware account and then follow steps described in *Add Administrators to Log in to Your Horizon Cloud Tenant Environment* using the row in the General Setup section of the screen.

5 Deploy the Horizon Cloud Connector virtual appliance into the pod's environment. Follow the OVA deployment steps described in *Download and Deploy the Horizon Cloud Connector into the Pod's vSphere Environment*.

6 After the virtual appliance is powered up, enable SSH access to the virtual appliance to remotely run commands in the appliance's operating system. You must use your vSphere environment to launch the appliance's console, log in to the appliance's operating system using the root user that you set during deployment of the appliance, and run the appropriate commands to enable SSH. Follow the steps in *Enable SSH Access to the Horizon Cloud Connector Prior to Pairing it with Your Horizon Connection Server*.

   **Note** Those steps to enable SSH are used for the time when the pod is not yet successfully paired with Horizon Cloud. After the pod is successfully paired with Horizon Cloud, then you will be able to use the browser-based Horizon Cloud Connector's configuration portal to enable and disable SSH access to the virtual appliance.

7 If your environment requires use of a proxy and you did not specify proxy-related settings in the OVF deployment wizard, configure the proxy-related settings for the virtual appliance. See *Modifying Proxy Settings for Horizon Cloud Connector 1.5 or Earlier* for more details.

8 If you want to access the browser-based Horizon Cloud Connector configuration portal using a fully-qualified domain name (FQDN) instead of using the Horizon Cloud Connector virtual appliance's IP address, create a forward and reverse lookup record in your DNS server that maps an FQDN to the virtual appliance's IP address.
9 (Horizon Cloud Connector 1.6 or later) Verify the health of the pod's system components and services by opening an SSH session to the Horizon Cloud Connector virtual appliance and running the precheck.sh diagnostics script. See Verify the Horizon Pod and Virtual Appliance Are Ready to Be Paired for more details.

10 Using either a mapped FQDN or the virtual appliance's IP address, log in to the browser-based Horizon Cloud Connector configuration portal and complete the onboarding steps that pair the connector with the pod's Connection Server. Follow the steps described in Complete Pairing the Horizon Pod with Horizon Cloud Using the Horizon Cloud Connector Configuration Portal.

**Tip** When the connector and Connection Server are successfully paired, the Horizon Cloud Connector configuration portal will display a Congratulations message. At this point, VMware will activate your subscription license. The activation is typically completed after 30 minutes but can take up to 48 hours in some cases. When the license is activated, you will see the message Connected to License Service in your pod's web-based console's **Product Licensing & Usage** screen.

11 Depending on your team standard practices and environment, optionally configure the Horizon Cloud Connector virtual appliance in areas such as configuring a CA-signed certificate and setting a password expiry for the appliance's root user. See the topic.

12 Complete the Active Directory domain registration workflow within the Horizon Cloud administrative console, named Horizon Cloud Administration Console, or console for short. See the topic Typical Administrative and Maintenance Tasks You Perform on the Horizon Cloud Connector After the Horizon Pod is Paired with Horizon Cloud in the Administration Guide.

**Tip** Completing the Active Directory domain registration workflow enables you to take advantage of all of the cloud-hosted services, such as the Cloud Monitoring Service (CMS). Until the pod's Active directory domain is registered with your tenant environment, the console's screens in which the CMS's monitoring data gets displayed are inaccessible.

13 Give the Horizon Cloud Super Administrators role to an Active Directory group that includes that domain-join account as a member. See the topic Assign Horizon Cloud Administrative Roles to Active Directory Groups in the Administration Guide.
You can find in-depth details on how to accomplish each workflow step in the topics that are linked from each step above or in the companion guide. See the Horizon Cloud Administration Guide.

DNS, Ports, and Protocols Requirements When Using Horizon Cloud Connector and a Horizon Pod

When you are using the Horizon Cloud Connector virtual appliance with your Horizon pod, you must configure your firewalls to allow the appliance to access the Domain Name Service (DNS) addresses it needs. In addition, your proxy settings require configured ports and protocols and DNS must resolve specific names as described in this topic. Then, after the Horizon Cloud Connector virtual appliance is deployed and you have completed the steps to successfully connect the pod to Horizon Cloud, specific ports and protocols are required for ongoing operations between Horizon Cloud and the virtual appliance.

As described in When Onboarding a Horizon Pod to Use Horizon Subscription Licenses or Cloud-Hosted Services with that Pod, the Horizon Cloud Connector virtual appliance is used with VMware Horizon deployments to activate subscription licenses on Horizon and enable use of cloud-hosted services with your Horizon deployments.

Connectivity and DNS Requirements

The steps for connecting Horizon Cloud with your Horizon pod using the Horizon Cloud Connector include the step to use a browser to navigate to the Horizon Cloud Connector appliance's IP address and a login screen will appear. To see that login screen requires Internet connectivity between the Horizon Cloud Connector appliance and the Horizon Cloud cloud control plane. The appliance establishes a connection to the Horizon Cloud cloud control plane initially using HTTPS, and then opens a persistent WebSocket connection, using outbound Internet port 443. For ongoing operations, the connection between the Horizon Cloud Connector appliance and Horizon Cloud requires that outbound Internet connection using port 443 open all the time. You must ensure the following DNS names are resolvable and reachable using the specific ports and protocols as listed according to the tables below.

Note If you are going to enable use of Universal Broker with the pod, there might be additional connectivity requirements for that use case. For details, see System Requirements for Universal Broker and its related topics.

Your Welcome to Horizon Service email will indicate which regional control plane instance your tenant account was created in. Due to a known issue that existed when the welcome email was sent to you, the email you received might display the system string names used for the regions instead of human-friendly names. If you see a system string name in your welcome email, you can use the following table to relate what is shown in your email with the regional control plane DNS names.
## Table 2-1. Regions in Your Welcome Email Mapped to Regional Control Plane DNS Names

<table>
<thead>
<tr>
<th>Your welcome email says</th>
<th>Regional DNS Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>USA</td>
<td>cloud.horizon.vmware.com</td>
</tr>
<tr>
<td>EU_CENTRAL_1 or Europe</td>
<td>cloud-eu-central-1.horizon.vmware.com</td>
</tr>
<tr>
<td>AP_SOUTHEAST_2 or Australia</td>
<td>cloud-ap-southeast-2.horizon.vmware.com</td>
</tr>
<tr>
<td>PROD1_NORTHCENTRALUS2_CP1 or USA-2</td>
<td>cloud-us-2.horizon.vmware.com</td>
</tr>
<tr>
<td>PROD1_NORTHEUROPE_CP1 or Europe-2</td>
<td>cloud-eu-2.horizon.vmware.com</td>
</tr>
<tr>
<td>PROD1_AUSTRALIAEAST_CP1 or Australia-2</td>
<td>cloud-ap-2.horizon.vmware.com</td>
</tr>
<tr>
<td>Japan</td>
<td>cloud-jp.horizon.vmware.com</td>
</tr>
<tr>
<td>Source</td>
<td>Destination (DNS name)</td>
</tr>
<tr>
<td>------------------------</td>
<td>---------------------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| Horizon Cloud Connector | One of the following names, depending on which regional Horizon Cloud control plane instance is specified in your Horizon Cloud tenant account. The regional instance is set when the account is created, as described in Onboarding to Horizon Cloud for Microsoft Azure, Horizon On-Premises, and Horizon on VMware Cloud on AWS.  
- cloud.horizon.vmware.com  
- cloud-us-2.horizon.vmware.com  
- cloud-eu-central-1.horizon.vmware.com  
- cloud-eu-2.horizon.vmware.com  
- cloud-ap-southeast-2.horizon.vmware.com  
- cloud-ap-2.horizon.vmware.com  
- cloud-jp.horizon.vmware.com | 443 | TCP       | Regional Horizon Cloud control plane instance  
- United States: cloud.horizon.vmware.com, cloud-us-2.horizon.vmware.com  
- Europe: cloud-eu-central-1.horizon.vmware.com, cloud-eu-2.horizon.vmware.com  
- Asia Pacific: cloud-ap-southeast-2.horizon.vmware.com, cloud-ap-2.horizon.vmware.com  
- Japan: cloud-jp.horizon.vmware.com | 443 | TCP       | Cloud Monitoring Service (CMS)                   |

### Ports and Protocols Required by the Horizon Cloud Connector Virtual Appliance

For ongoing operations between Horizon Cloud Connector and Horizon Cloud, the ports and protocols in the following table are required.
Table 2-2. Horizon Cloud Connector Ports

<table>
<thead>
<tr>
<th>Source</th>
<th>Target</th>
<th>Port</th>
<th>Protocol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Horizon Cloud Connector</td>
<td>Horizon Cloud</td>
<td>443</td>
<td>HTTPS</td>
<td>Used to pair the Horizon Cloud Connector with Horizon Cloud and transfer data.</td>
</tr>
<tr>
<td>Horizon Cloud Connector</td>
<td>Connection Server</td>
<td>443</td>
<td>HTTPS</td>
<td>API calls to Connection Server.</td>
</tr>
<tr>
<td>Horizon Cloud Connector</td>
<td>Connection Server</td>
<td>4002</td>
<td>TCP</td>
<td>Java Message Service (JMS) communication between the Cloud Connector and the Connection Server</td>
</tr>
<tr>
<td>New version of the Horizon Cloud Connector appliance</td>
<td>Existing version of the Horizon Cloud Connector appliance</td>
<td>22</td>
<td>SSH</td>
<td>Listen for requests to start the update process.</td>
</tr>
<tr>
<td>Web browser</td>
<td>Horizon Cloud Connector</td>
<td>443</td>
<td>HTTPS</td>
<td>Listen for the initiation of the pairing process.</td>
</tr>
<tr>
<td>Horizon Cloud Connector</td>
<td>Certificate Authority</td>
<td>*</td>
<td>HTTP, HTTPS</td>
<td>CRL or OCSP queries</td>
</tr>
<tr>
<td>Cloud Monitoring Service agent in the desktop or server VMS that are from the cloud-connected Horizon pod on your network</td>
<td>Horizon Cloud Connector appliance</td>
<td>11002</td>
<td>TCP</td>
<td>For the Cloud Monitoring Service agent on a server or desktop VM to send data to the Horizon Cloud Connector</td>
</tr>
</tbody>
</table>

Preparing to Run the Onboarding Wizard to Pair a Horizon Pod with Horizon Cloud Using Horizon Cloud Connector

To connect an existing Horizon pod with the current service release level for the purposes of using Horizon subscription licenses or cloud-hosted services with that pod, ensure you have the following items in place before you deploy the Horizon Cloud Connector appliance and running the onboarding wizard.

- Verify you have met the prerequisites described in VMware Horizon Pods with Horizon Cloud - Requirements Checklist - Updated As Appropriate for Connecting Pods Starting From the July 2020 Service Release.
- You have determined which of the pod’s Connection Server instances you will pair with the Horizon Cloud Connector virtual appliance and you have the FQDN of that Connection Server instance. You can pair the Horizon Cloud Connector virtual appliance with only one of the pod’s installed Connection Server instances at any given time.
- You have determined which of the pod’s administrator accounts you will specify when pairing the Connection Server with the Horizon Cloud Connector and that administrator account meets the requirements needed for pairing. This Active Directory user must have the Horizon
predefined Administrators role on the root access group, as displayed in your pod’s web-based console in Global Administrators View > Role Permissions > Administrators. In other words, the Active Directory user specified for the pod-onboarding process is a super user for that pod, as described in the Horizon documentation's Horizon Console Administration guide.

- You have a valid My VMware account at https://my.vmware.com. This account is required to purchase a Horizon subscription license, for running the Horizon Cloud Connector onboarding workflow to pair the pod with the cloud service, and for logging in to the cloud-based administrative console to use the cloud-hosted services with that pod.

- A Horizon subscription license is associated with that My VMware account.

- You have the information from the welcome email that was sent to that My VMware account’s associated email address, including the link for downloading the Horizon Cloud Connector OVA. If you do not have the email information, you can download the OVA by logging in to my.vmware.com using that My VMware account and navigating to the product downloads that are listed for that account.

- If you are using the Microsoft Internet Explorer Web browser, verify that the compatibility mode is off. This setting enables viewing the Horizon Cloud Connector appliance onboarding user interface in that web browser.

- Verify that you have met the DNS, Ports, and Protocols Requirements When Using Horizon Cloud Connector and a Horizon Pod.

- Determine the static IP address that you will use for the Horizon Cloud Connector virtual appliance. You will need this IP address when you deploy the Horizon Cloud Connector OVA in vSphere.

**Note** Do not use IPv6 with the Horizon Cloud Connector virtual appliance. IPv6 is not supported.
Verify you have the typical networking information that is appropriate for your environment to use when you deploy the Horizon Cloud Connector OVA in your vSphere environment, such as your DNS search domain, DNS server IP address, default gateway address, and subnet mask.

**Note** Proxy SSL configuration is not supported for the self-signed certificate for the Horizon Cloud Connector virtual appliance.

**Important** If you are deploying Horizon Cloud Connector 1.5 or earlier and your environment requires use of a proxy server with the appliance, review the information described in **Modifying Proxy Settings for Horizon Cloud Connector 1.5 or Earlier** before deploying the appliance. In addition, keep the following known limitations in mind. These limitations do not apply to Horizon Cloud Connector 1.6 or later.

- Due to a known limitation, any no-proxy host configuration that you specify when deploying the OVF template is not honored by the system. To configure a no-proxy host configuration for Horizon Cloud Connector 1.5 or earlier, you must change a configuration file in the appliance after it is deployed. For details, see [VMware Knowledge Base article 76663](https://kb.vmware.com/kb/76663).

- If you plan to use the pod with Universal Broker and Horizon Cloud Connector 1.5, and your environment requires using proxy settings, you must configure those proxy settings when you deploy the OVF template. Due to a known limitation of Horizon Cloud Connector 1.5, Universal Broker will not honor any modifications made to proxy settings after deployment. Since you can only configure no-proxy hosts after deployment, this limitation means that Universal Broker does not support the use of no-proxy hosts with Horizon Cloud Connector 1.5.

- Verify you have a strong root password for the Horizon Cloud Connector virtual appliance containing a minimum of eight characters with one capital, one numeric, and one special character.

**Important** When deploying the OVF template, you must specify a root password that meets the security standards of a strong password. However, due to a known limitation, the OVF deployment wizard continues to deploy the virtual appliance even if you specify a root password that does not contain a special character. In this case, the deployment succeeds but you will be blocked from logging in to the virtual appliance’s operating system after deployment.

To ensure your access to the virtual appliance after it is deployed, always specify a strong root password containing at least one special character as prompted by the OVF deployment wizard.
For the minimum use case of using Horizon subscription licenses with the Horizon pod, verify that you meet these prerequisites, in addition to the ones listed above:

- The Connection Server instance with which you will be pairing the Horizon Cloud Connector must be running version 7.6 or later. Version 7.6 is the minimum version that can be paired with the cloud service.

**Tip**  Even though technically you can pair a Horizon pod that is running versions older than the latest version, to obtain the most up-to-date cloud-hosted features with that pod, the pod’s Connection Servers would best be running the most current software version. Only by using the most current combination of the Connection Server version and Horizon Cloud Connector version will you get access to the most up-to-date features of the cloud-hosted features beyond simply using a subscription license with that pod.

### Horizon Cloud Connector Known Considerations

Keep these considerations in mind when you are using Horizon Cloud Connector.

- You must deploy the Horizon Cloud Connector virtual appliance into your pod’s vSphere environment using vSphere Client or vSphere Web Client. Do not deploy the appliance directly into the ESXi host.

- Use of IPv6 with the Horizon Cloud Connector virtual appliance is not supported.

- Proxy SSL configuration is not supported during the deployment of the Horizon Cloud Connector virtual appliance.

- Due to a known limitation, any no-proxy host configuration that you specify when deploying the OVF template is not honored by Horizon Cloud Connector 1.5 or earlier. To configure a no-proxy host configuration, you must change a configuration file in the appliance after it is deployed. For details, see VMware Knowledge Base article 76663. This limitation does not apply to Horizon Cloud Connector 1.6 or later.

- If you plan to use the pod with Universal Broker and Horizon Cloud Connector 1.5, and your environment requires using proxy settings, you must configure those proxy settings when you deploy the OVF template. Due to a known limitation, Universal Broker will not honor any modifications made to proxy settings after deployment. Since you can only configure no-proxy hosts after deployment, this limitation means that the use of no-proxy hosts with Universal Broker is not supported by Horizon Cloud Connector 1.5. This limitation does not apply when you use Universal Broker with Horizon Cloud Connector 1.6 or later.

- Information about the static IP and proxy settings for the deployed Horizon Cloud Connector virtual appliance is saved in certain container files. When you want to change those settings on the virtual appliance, you must connect to the virtual appliance and edit those container files. If you change the static IP address for the deployed virtual appliance in your vSphere environment, you must edit the appropriate container file in the virtual appliance's operating system and run a command to ensure that the new IP address is shared with all the pod’s components that depend upon the virtual appliance. See Update the Static IP for the Horizon Cloud Connector Virtual Appliance.
Before deleting the Horizon Cloud Connector virtual appliance from your vCenter environment, point your browser to the Horizon Cloud Connector appliance's IP address and use the Unplug action to remove the connection between the pod and Horizon Cloud.

Using a separate vdmadmin account for the Horizon Cloud Connector paired with the Horizon pod is a best practice. Using a separate vdmadmin account avoids configurations being overridden between cloud and on-premises management. Using separate accounts also provides easier auditing for the cloud-based operations.

The connection between the Horizon Cloud Connector and Horizon Cloud uses outbound Internet port 443. For all the connector's required DNS, ports, and protocols, see DNS, Ports, and Protocols Requirements When Using Horizon Cloud Connector and a Horizon Pod.

You set the password for the root user of the Horizon Cloud Connector virtual appliance during deployment. By default, this password does not expire. However, based on your organization's security policy, you might want to periodically update that root password by setting an expiry policy for that root user. For steps, see Set a Password Expiry Policy for the Horizon Cloud Connector Root User.

If your Connection Server is using self-signed certificates and you then replace those self-signed certificates after pairing the pod with Horizon Cloud, you must log in to the Horizon Cloud Connector interface and use the Reconfigure workflow to perform the certificate validation steps again with the new self-signed certificate. When you log in to the Horizon Cloud Connector interface, you can click Reconfigure and complete the wizard steps to verify communication using the new self-signed certificate from the Connection Server.

Similarly, upgrading your Connection Server might result in changes to self-signed certificates. To ensure validation of the new certificates, perform the Reconfigure workflow for the Horizon Cloud Connector after upgrading your Connection Server.

If you added an entry to the /etc/hosts file to resolve the IP address of the Connection Server, you must restart the hze-core and csms services. Use the following commands:

```
systemctl restart hze-core
systemctl restart csms
```

To ensure that the Horizon Cloud Connector virtual appliance authenticates correctly with Horizon Cloud and the required Connection Server instances, you must synchronize the virtual appliance's clock with an NTP server. For more information, see Synchronize the Horizon Cloud Connector Virtual Appliance with an NTP Server.

If you encounter connectivity issues with the Horizon Cloud Connector configuration portal, see the troubleshooting information in VMware Knowledge Base (KB) article VMware Knowledge Base (KB) article 79859.
Connect Horizon Cloud Service with an Existing Horizon Pod to Use Horizon Subscription Licenses or Cloud-Hosted Services or Both

Connecting your existing Horizon pod with Horizon Cloud Service is a multi-step process. After you purchase a VMware Horizon subscription license, VMware sends you a license subscription email, which includes the link to download the Horizon Cloud Connector virtual appliance. Then you install the virtual appliance and power it on. After the virtual appliance is powered on and you verify the health of the required pod components and services, you use the connector’s onboarding workflow to pair it with a Connection Server in the pod with which you want to use that subscription license. As part of a successful pairing process, the Horizon Cloud Connector virtual appliance bridges the Connection Server to Horizon Cloud Service, so that the cloud management plane can manage the Horizon subscription license and other cloud-hosted services for that now cloud-connected pod.

Tip  For an introduction to the overall Horizon Cloud onboarding process, see Onboarding to Horizon Cloud for Microsoft Azure, Horizon On-Premises, and Horizon on VMware Cloud on AWS.

You use the steps here to connect Horizon pods that already exist, installed on-premises or installed in VMware Cloud™ on AWS, to the cloud management plane. Connecting a pod to Horizon Cloud Service allows you to use Horizon subscription licenses with that pod, even if you do not take advantage of any cloud-hosted services with that pod. With a Horizon subscription license, you do not need to manually enter a license key to activate the license for the pod. After the pairing is complete, VMware activates the subscription license, typically within 48 hours after you paired the pod with the cloud control plane. When VMware has activated the subscription license, a message appears in the pod’s web-based management console that says your Horizon environment is using the subscription license.
As described in the Onboarding to Horizon Cloud for Microsoft Azure, Horizon On-Premises, and Horizon on VMware Cloud on AWS, the process to onboard a Horizon pod to Horizon Cloud involves these basic concepts:

- Horizon subscription licenses are managed from the cloud management plane, which is Horizon Cloud.

- Therefore, if you want to use subscription licenses with a Horizon pod, you must connect the pod with that cloud management plane. If you wish to avoid connecting that pod with the cloud management plane, you will not be able to use subscription licenses with that pod.

- Connecting that existing Horizon pod with the cloud management plane requires a connector, which is named Horizon Cloud Connector. The cloud management plane talks to the connector, which in turn talks to one of the pod’s Connection Server instances. The connector is paired with only one of the pod’s installed Connection Server instances at any given time.

- Because the Horizon Cloud Connector has to reach both the cloud management plane and the pod’s Connection Server instance that you are pairing it with, specific DNS, Ports, and Protocols Requirements When Using Horizon Cloud Connector and a Horizon Pod must be met to have a successful outcome of pairing the Horizon Cloud Connector with the pod and for ongoing operations. Even the minimal use case of using a subscription license with the pod requires meeting those DNS, ports, and protocols requirements.

- A My VMware account is required to obtain a Horizon subscription license and to authenticate with the cloud management plane to set up the connector and establish the connection to use that subscription license with the pod.

- You might want to use only subscription licenses with your Horizon pod, or you might want to additionally use cloud-hosted services with that pod. Both of those use cases require a subscription license.

- To obtain the most up-to-date features and security and bug fixes, you would best use the combination of running the most current version of the Connection Server software with the most current version of Horizon Cloud Connector. Even though version 7.6 was the first Horizon version that provided the ability to use a Horizon subscription license managed from the cloud management plane using the Horizon Cloud Connector, that version is over a year
old now and none of the current cloud management services beyond the simple use of the subscription license are available for that version 7.6. For the supported combinations of Horizon software versions and Horizon Cloud Connector versions, see VMware Knowledge Base article 77584.

The high-level steps of this process are:

1. Obtain your My VMware account.
2. Use that My VMware account to sign up for a Horizon subscription license.
3. When you sign up for the license, a welcome email is sent to the email address associated with that My VMware account. That welcome email will contain a link to download the Horizon Cloud Connector OVA from the appropriate my.vmware.com Downloads area.
4. Download the Horizon Cloud Connector OVA using the link in the welcome email.
5. Deploy that OVA into the pod's vSphere environment using a static IP address. When the virtual appliance deployment process is completed, power on the virtual appliance.
6. When powered on, the appliance's blue console screen will display a URL address to use to start the pairing workflow to pair the Horizon Cloud Connector with a Connection Server instance in the pod and complete the connection between the Connection Server instance, the Horizon Cloud Connector, and the cloud management plane.
7. (Horizon Cloud Connector 1.6 or later) Before starting the pairing workflow, run the precheck.sh script to verify the health of the pod's system components and services.
8. Use the URL address that was displayed on the appliance's blue console screen in a browser to start the pairing workflow. The login screen from Horizon Cloud will display and you log in using the My VMware account credentials. At that point, the workflow's user interface appears in the browser and you complete the steps as detailed in this topic below.

**Important** If you already have cloud-connected pods in your Horizon Cloud environment to which you are connecting this pod, all those cloud-connected pods must have line of sight to the same set of Active Directory domains. When performing the steps to connect the not-yet-connected pod, you must ensure that the pod will have a line of sight to those Active Directory domains that are already registered with your Horizon Cloud environment.

As an example, if your environment already has pods in Microsoft Azure and you are connecting a Horizon pod, you must ensure:

- The Horizon pod you are connecting using the following steps has a line of sight to the Active Directory domains used by those existing pods in Microsoft Azure, because those domains are already registered with the cloud plane for your environment.
- Your existing cloud-connected pods in Microsoft Azure have line of sight to the Horizon pod's Active Directory domain, the domain you are using in the following steps to pair the Horizon Cloud Connector virtual appliance with the Horizon pod's Connection Server.
Prerequisites

Verify that you have fulfilled all of the items described in Preparing to Run the Onboarding Wizard to Pair a Horizon Pod with Horizon Cloud Using Horizon Cloud Connector.

Verify that you have met the DNS, Ports, and Protocols Requirements When Using Horizon Cloud Connector and a Horizon Pod when using the Horizon Cloud Connector to pair a Horizon pod with Horizon Cloud.

Verify that your DNS configuration in your network topology will provide for the deployed Horizon Cloud Connector to resolve the FQDN of the pod’s Connection Server. If the deployed Horizon Cloud Connector cannot resolve the Connection Server using DNS, the onboarding wizard will encounter an unexpected error at the step where you enter the pod’s domain credentials.

Review Horizon Cloud Connector Known Considerations to ensure you are aware of those items.

The Horizon Cloud Connector virtual appliance must reach the Internet to talk to the Horizon Cloud control plane. If your environment requires use of a proxy server and proxy configuration for virtual appliances to reach the Internet, verify that you have reviewed the proxy-related information, known limitations, and known issues when using proxy settings with the Horizon Cloud Connector appliance. See the proxy-related information in Preparing to Run the Onboarding Wizard to Pair a Horizon Pod with Horizon Cloud Using Horizon Cloud Connector, Horizon Cloud Connector Known Considerations, and Modifying Proxy Settings for Horizon Cloud Connector 1.5 or Earlier.

Important If you plan to use the pod with Universal Broker and Horizon Cloud Connector 1.5, and your environment requires using proxy settings, you must configure those proxy settings when you deploy the OVF template. Due to a known limitation of Horizon Cloud Connector 1.5, the Universal Broker will only honor proxy settings that are specified at OVF template deployment. As a result, due to another known issue in Horizon Cloud Connector 1.5, where any no-proxy host configuration that is specified in the OVF template deployment is not honored by the appliance and must be configured post-deployment, this known limitation that the combination of Universal Broker and Horizon Cloud Connector 1.5 requires configuring proxy settings during OVF template deployment means use of no-proxy hosts with Universal Broker is not currently supported.

Procedure

1 Download and Deploy the Horizon Cloud Connector into the Pod’s vSphere Environment

In this step in the workflow of onboarding a Horizon pod to Horizon Cloud, you download and deploy the Horizon Cloud Connector in your vSphere environment. The result of these steps is the Horizon Cloud Connector virtual appliance is deployed and running in your vSphere environment.
2 Verify the Horizon Pod and Virtual Appliance Are Ready to Be Paired

In this step in the workflow of onboarding a Horizon pod to Horizon Cloud, you run the precheck.sh diagnostic tool to validate that the pod and Horizon Cloud Connector are both ready for the pairing process. By first running the diagnostics and remediating any blocking issues found in system components and configurations, you can maximize your chances of success when pairing the pod with Horizon Cloud.

3 Complete Pairing the Horizon Pod with Horizon Cloud Using the Horizon Cloud Connector Configuration Portal

In this step in the workflow of onboarding a Horizon pod to Horizon Cloud, you use the Horizon Cloud Connector configuration portal to specify details that the Horizon Cloud Connector uses to pair with the Horizon pod’s Connection Server. Completing these steps successfully results in that pod connected to your Horizon Cloud tenant environment.

Results

When the Horizon pod is successfully connected with Horizon Cloud, the Horizon Cloud Connector configuration portal displays a Congratulations message. From this point, you use this same configuration portal to perform administrative tasks such as review the health status of the Horizon Cloud Connector components, enable or disable SSH access to the Horizon Cloud Connector virtual appliance, and similar tasks. For details, see the topic Typical Administrative and Maintenance Tasks You Perform on the Horizon Cloud Connector After the Horizon Pod is Paired with Horizon Cloud.

What to do next

If your only goal is to use your subscription license with the now cloud-connected pod, there are no additional steps to take, except to ensure that the DNS, ports, and protocols requirements continue to be met to maintain the connection between Horizon Cloud Connector and the cloud control plane. Because the subscription licenses are managed by the cloud control plane, Horizon Cloud Connector must continue to reach the cloud control plane for the pod to receive your subscription license information.

Important When the Horizon Cloud Connector is installed, the connection is established to the cloud control plane by outbound Internet port 443. This connection will be open all the time. If this connection between the Cloud Connector and the cloud control plane goes offline, there is a grace period of 10 days by default that can elapse before the pairing between this pod and Horizon Cloud is marked expired. If this happens, contact VMware Support for assistance.

If you want to leverage any of the cloud-hosted services with the cloud-connected pod, you must log in to the administrative console and complete the Active Directory registration workflow that registers the pod’s Active Directory domain with Horizon Cloud. For details of that workflow, see Performing Your First Active Directory Domain Registration in the Horizon Cloud Environment in the Horizon Cloud Administration Guide.
Download and Deploy the Horizon Cloud Connector into the Pod's vSphere Environment

In this step in the workflow of onboarding a Horizon pod to Horizon Cloud, you download and deploy the Horizon Cloud Connector in your vSphere environment. The result of these steps is the Horizon Cloud Connector virtual appliance is deployed and running in your vSphere environment.

**Important** You must deploy the Horizon Cloud Connector virtual appliance into your pod's vSphere environment using vSphere Client or vSphere Web Client. Do not deploy the appliance directly into the ESXi host.

**Prerequisites**

- Verify that you have met the connector-related prerequisites described in Preparing to Run the Onboarding Wizard to Pair a Horizon Pod with Horizon Cloud Using Horizon Cloud Connector.

- Verify that you have met the DNS, Ports, and Protocols Requirements When Using Horizon Cloud Connector and a Horizon Pod for using Horizon Cloud Connector to pair a Horizon pod with Horizon Cloud.

- The Horizon Cloud Connector virtual appliance must reach the Internet to talk to the Horizon Cloud control plane. If your environment requires use of a proxy server and proxy configuration for deployed appliances to reach the Internet, verify that you have reviewed the proxy-related information, known limitations, and known issues when using proxy settings with the Horizon Cloud Connector appliance. See the proxy-related information in Preparing to Run the Onboarding Wizard to Pair a Horizon Pod with Horizon Cloud Using Horizon Cloud Connector, Horizon Cloud Connector Known Considerations, and Modifying Proxy Settings for Horizon Cloud Connector 1.5 or Earlier.

**Procedure**

1. As described in the prerequisites list, download the Horizon Cloud Connector appliance using the link provided in your subscription email.

   The Horizon Cloud Connector appliance is available as an OVA file and has its home location in my.vmware.com after you log in to my.vmware.com using your My VMware account credentials.

   **Important** Ensure that the downloaded version is the most recent version —version 1.7 or later — to enable the most up-to-date features. If you previously downloaded a Horizon Cloud Connector OVA with a version prior to 1.7, log in to my.vmware.com and obtain the most recent version to use to pair your pod.

2. Use vSphere Client or vSphere Web Client to deploy the Horizon Cloud Connector appliance as an OVF template into your Horizon pod.

   For general information about deploying OVF templates, see the vSphere Virtual Machine Administration guide on the VMware vSphere Documentation page.
The OVF deployment wizard has several steps, where you make typical OVF deployment choices such as which host, which datastore, which network, and so on. The **Customize template step** is where you provide details that are specific for the Horizon Cloud Connector appliance.

**Important** You might have best results deploying the appliance into your pod when you use the VMware vSphere® Web Client, which is a Flex-based user interface. If you deploy using the VMware vSphere® Client™, which is the newest HTML5-based user interface, you might encounter an error message about an invalid value. That issue is caused by a known issue in the vSphere Client, and is not an issue with the Horizon Cloud Connector appliance package. If you encounter that error, deploy the appliance using the Flex-based user interface instead.
3 In the **Customize template** wizard step, provide input for the required fields and for those items that are appropriate for your environment.

The input in this step is used to configure the virtual appliance.

a Specify a root password for the appliance that meets the security standards of a strong password. Verify that the password contains a minimum of eight characters with at least one capital, one numeric, and one special character.

**Important** Due to a known limitation, the OVF deployment wizard continues to deploy the virtual appliance even if you specify a root password that does not contain a special character. In this case, the deployment succeeds but you will be blocked from logging in to the virtual appliance's operating system after deployment. To ensure your access to the virtual appliance after it is deployed, verify that your root password contains at least one special character.

b Specify a static IP address for the virtual appliance.

Do not use IPv6 with the Horizon Cloud Connector virtual appliance. IPv6 is not supported.

c If your environment requires use of an HTTP proxy server for your virtual appliances to access the Internet, configure the proxy-related settings.

**Important** Keep the following considerations in mind:

- Proxy SSL configuration is not supported for the self-signed certificate for the Horizon Cloud Connector virtual appliance.

- (Horizon Cloud Connector 1.6 or later) To ensure that only outbound requests to the Internet route through the HTTP proxy, configure no-proxy hosts that bypass the proxy server when receiving internal requests from the appliance. At the minimum, for **No Proxy For**, enter the DNS subdomain of the Connection Server and vCenter Server instances associated with the pod that will be paired with Horizon Cloud Connector. You can also specify no-proxy hosts by entering an IP range, using a comma separator between entries as shown in the following example:

```
.ad-domain.example.com, 10.109.*
```

- (Horizon Cloud Connector 1.6 or later) If you leave the **No Proxy For** setting blank, the virtual appliance fetches the Connection Server host names provided by the administrator or discovered through querying the pod and configures those hosts as implicit no-proxy hosts.

- (Horizon Cloud Connector 1.5 or earlier) Due to a known limitation, any no-proxy host configuration that you specify when deploying the OVF template is not honored by the system. To configure a no-proxy host configuration, you must change a configuration file in the appliance after it is deployed as described in VMware Knowledge Base article 76663. This limitation does not apply to Horizon Cloud Connector 1.6 or later.

- (Horizon Cloud Connector 1.5) If you plan to use the pod with Universal Broker and Horizon Cloud Connector 1.5, you must configure all required proxy settings when you deploy the OVF template. Due to a known limitation of Horizon Cloud Connector 1.5, Universal Broker will not honor any modifications made to proxy settings after deployment.
4  Power on the Horizon Cloud Connector appliance.

5  When the appliance is fully powered on, use the vSphere Web Client's option to start the Horizon Cloud Connector appliance's console.

The Horizon Cloud Connector appliance's blue console screen appears. That blue console screen displays the URL address to load in your browser for the onboarding workflow. The following screenshot is an example for a deployed appliance that has address https://10.92.245.255/.

6  Complete the steps in Enable SSH Access to the Horizon Cloud Connector Prior to Pairing it with Your Horizon Connection Server.

7  If you want to use a fully qualified domain name (FQDN) for the Horizon Cloud Connector virtual appliance and resolve the host name, create a forward and reverse lookup record in your DNS server that maps that FQDN to the Horizon Cloud Connector virtual appliance's static IP.

8  Continue with the pod onboarding workflow by doing one of the following.

   - If you have deployed Horizon Cloud Connector 1.6 or later, proceed to Verify the Horizon Pod and Virtual Appliance Are Ready to Be Paired.
   - If you have deployed Horizon Cloud Connector 1.5 or earlier, proceed to Complete Pairing the Horizon Pod with Horizon Cloud Using the Horizon Cloud Connector Configuration Portal.

Enable SSH Access to the Horizon Cloud Connector Prior to Pairing it with Your Horizon Connection Server

Use these steps if you want to be able to use SSH with the deployed Horizon Cloud Connector appliance prior to running the onboarding wizard to pair the Horizon Cloud Connector with the pod.

Prerequisites

Verify the Horizon Cloud Connector appliance is successfully deployed in your vSphere environment as described in Connect Horizon Cloud Service with an Existing Horizon Pod to Use Horizon Subscription Licenses or Cloud-Hosted Services or Both, but not yet paired with the Connection Server.
Procedure

1. Use vSphere Web Client to launch the console for the deployed appliance and log in to the appliance using the root account and the password you set when you deployed the OVA into vSphere.

2. In the appliance's operating system, enable SSH by running the following command.

   ```bash
   /opt/vmware/bin/configure-adapter.py --sshEnable
   ```

Results

SSH access to the appliance is enabled.

To disable SSH access, use the following command:

```bash
/opt/vmware/bin/configure-adapter.py --sshDisable
```

What to do next

Perform the onboarding wizard steps to complete pairing the Horizon Cloud Connector with the pod. When pairing is successfully completed, the Horizon Cloud Connector web-based configuration portal will provide a toggle that you can use to disable SSH access for the appliance, or re-enable SSH if it was previously disabled.

Modifying Proxy Settings for Horizon Cloud Connector 1.6 or Later

You can configure HTTP proxy settings during deployment of the Horizon Cloud Connector OVF template. If you want to modify these proxy settings after deployment, you must use the `configure-webproxy.py` command. The `configure-webproxy.py` command is located in the `/opt/vmware/bin` directory of the deployed Horizon Cloud Connector appliance.

**Note**  
Observe the following guidelines with respect to proxy settings and appliance updates:

- If you manually update Horizon Cloud Connector 1.6 or later to a newer version, you must reconfigure your proxy settings. Your original proxy configuration does not carry over with the manual appliance update.

- If Horizon Cloud Connector 1.6 or later is automatically updated to a newer version, your proxy settings carry over with the automatic update. You do not need to reconfigure the proxy settings.

- To view the existing proxy settings for the Horizon Cloud Connector virtual appliance, run the following command.

  ```bash
  cat /opt/container-data/cc-settings/proxy.conf
  ```

Syntax for Using `configure-webproxy.py`

Use the following syntax to create a script with `configure-webproxy.py`:

```bash
configure-webproxy.py [argument1 [value1]] [argument2 [value2]] ...
```
To display the command usage and list of available arguments, run `configure-webproxy.py -h` or `configure-webproxy.py --help`.

**Arguments for configure-webproxy.py**

All arguments are optional for the `configure-webproxy.py` script.

<table>
<thead>
<tr>
<th>Argument</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>--proxyHost</code></td>
<td>Host name or IP address of the HTTP proxy server</td>
</tr>
<tr>
<td><code>--proxyPort</code></td>
<td>Port number for the proxy connection</td>
</tr>
<tr>
<td><code>--noProxyFor</code></td>
<td>Hosts or network range configured to bypass the HTTP proxy. Use spaces to separate multiple values.</td>
</tr>
<tr>
<td><code>--proxySsl</code></td>
<td>Specifies whether to use SSL for the proxy connection. Allowed values are true or false.</td>
</tr>
<tr>
<td><code>--proxyUsername</code></td>
<td>User name for the HTTP proxy</td>
</tr>
<tr>
<td><code>--proxyPassword</code></td>
<td>Password for the HTTP proxy</td>
</tr>
<tr>
<td><code>--implicitNonProxyHosts</code></td>
<td>Specifies whether to add the paired pod's Connection Server and vCenter Server implicitly to the list of hosts that bypass the HTTP proxy. Allowed values are true or false. The default is true. If your environment requires internal requests to the Connection Server and vCenter Server to route through the proxy, set this argument to false. In this case, only the hosts specified explicitly by <code>--noProxyFor</code> bypass the proxy.</td>
</tr>
</tbody>
</table>

**Example Script**

```
configure-webproxy.py --proxyHost PROXYEXAMPLE --proxyPort 80 --proxySsl=false
--noProxyFor ".AD-DOMAIN.EXAMPLE.COM 10.109.*"
```

This example script configures the following proxy settings:

- PROXYEXAMPLE is the proxy server.
- The proxy connection uses port 80.
- The proxy connection does not use SSL.
- Hosts that fall under .AD-DOMAIN.EXAMPLE.COM and 10.109.* bypass the proxy.
- Also, the paired pod's Connection Server and vCenter Server implicitly bypass the proxy by default.

**Modifying Proxy Settings for Horizon Cloud Connector 1.5 or Earlier**

You can configure HTTP proxy settings during deployment of the Horizon Cloud Connector OVF template. If you want to change the proxy settings after deployment or configure no-proxy hosts, you must modify certain configuration files. Due to a known limitation, Horizon Cloud Connector 1.5 or earlier does not honor the no-proxy host configuration specified during deployment.
deployment. To configure no-proxy hosts, you must modify a certain configuration file after deployment.

**Important** Due to a known limitation, if you plan to use Universal Broker with Horizon Cloud Connector 1.5, and your environment requires using proxy settings, you must configure those proxy settings when you deploy the OVF template. Universal Broker does not recognize any proxy settings configured after deployment. Since you can only configure no-proxy hosts after deployment, this limitation means that the use of no-proxy hosts with Universal Broker is not supported by Horizon Cloud Connector 1.5.

### Configuring No-Proxy Hosts for Horizon Cloud Connector 1.5 or Earlier

During deployment of the Horizon Cloud Connector OVF template, the deployment wizard provides prompts for configuring no-proxy hosts. However, due to a known issue, Horizon Cloud Connector 1.5 or earlier does not honor the no-proxy host configuration specified during deployment. Instead, you must modify a certain configuration file after deployment to set up no-proxy hosts.

To ensure that only outbound requests to the Internet route through the HTTP proxy, configure no-proxy hosts that bypass the proxy server when receiving internal requests from the appliance. At the minimum, configure the Connection Server and vCenter Server instances of the paired pod to be no-proxy hosts.

**Note** If you update Horizon Cloud Connector 1.5 or earlier to a newer version, you must reconfigure your no-proxy hosts. Your original no-proxy host configuration does not carry over with the appliance update.

For detailed information about configuring no-proxy hosts after deployment of the virtual appliance, see VMware Knowledge Base (KB) article 76663: [Proxy Configuration issues and remediation for Horizon Cloud Connector](https://kb.vmware.com/s/article/76663).

### Modifying the Proxy Settings for Horizon Cloud Connector 1.5 or 1.4

Use the following steps to modify HTTP proxy settings for Horizon Cloud Connector 1.5 or 1.4 after deploying the appliance.

1. Open a Secure Shell (SSH) session to your deployed Horizon Cloud Connector virtual appliance.

2. Change the proxy details as needed in the following files:
   - `/opt/container-data/cc-settings/proxy.conf`
   - `/opt/container-data/data/hze-core/properties/hydra.properties`
   - `/opt/container-data/data/hze-ccc/config/ccc-core/sn.config`

3. Restart the necessary services.

```
systemctl restart hze-core
systemctl restart hze-ccc
systemctl restart csms
```
Modifying the Proxy Settings for Horizon Cloud Connector 1.3 or Earlier

Use the following steps to modify HTTP proxy settings for Horizon Cloud Connector 1.3 or earlier after deploying the appliance.

1. Open a Secure Shell (SSH) session to your deployed Horizon Cloud Connector virtual appliance.

2. Change the proxy details as needed in the following files:
   - /opt/vmware/var/lib/tomcat8/properties/hydra.properties
   - /opt/vmware/var/lib/tomcat8/properties/sn.config

3. Restart the necessary services.

   ```
   systemctl restart tomcat8
   systemctl restart cccService
   ```

Configure a CA-Signed Certificate for the Horizon Cloud Connector Virtual Appliance

For enhanced security, you can configure a custom CA-signed certificate for the Horizon Cloud Connector virtual appliance.

Prerequisites

- Verify that the full certificate chain is available in the PEM format.
- Verify that the private key is available in the PEM format.
- Verify that the FQDN and Subject Alt Name is included in the issued certificate.

Procedure

1. Open an SSH session to your deployed Horizon Cloud Connector virtual appliance.

2. Copy the CA-signed certificate in the directory /root/server.crt.

3. Copy the CA-signed key in the directory /root/server.key.

4. Back up the existing certificate.

   - (Horizon Cloud Connector version 1.4 or later) Use the following command:
     ```
     cp /opt/container-data/certs/hze-nginx/server.crt /opt/container-data/certs/hze-nginx/server.crt.orig
     ```

   - (Horizon Cloud Connector version 1.3 or earlier) Use the following command:
     ```
     cp /etc/nginx/ssl/server.crt /etc/nginx/ssl/server.crt.orig
     ```
5 Back up the existing key.

- (Horizon Cloud Connector version 1.4 or later) Use the following command:

```
cp /opt/container-data/certs/hze-nginx/server.key /opt/container-data/certs/hze-nginx/server.key.orig
```

- (Horizon Cloud Connector version 1.3 or earlier) Use the following command:

```
cp /etc/nginx/ssl/server.key /etc/nginx/ssl/server.key.orig
```

6 Copy the existing nginx.conf file.

- (Horizon Cloud Connector version 1.4 or later) Use the following command:

```
cp /opt/container-data/conf/hze-nginx/nginx.conf /opt/container-data/conf/hze-nginx/nginx.conf.orig
```

- (Horizon Cloud Connector version 1.3 or earlier) Use the following command:

```
cp /etc/nginx/nginx.conf /etc/nginx/nginx.conf.orig
```

7 Copy the CA certificate in the appropriate directory for your virtual appliance version.

- (Horizon Cloud Connector version 1.4 or later) Use the following command:

```
cp /root/server.crt /opt/container-data/certs/hze-nginx/server.crt
```

- (Horizon Cloud Connector version 1.3 or earlier) Use the following command:

```
cp /root/server.crt /etc/nginx/ssl/server.crt
```

8 Copy the CA certificate key file in the appropriate directory for your virtual appliance version.

- (Horizon Cloud Connector version 1.4 or later) Use the following command:

```
cp /root/server.key /opt/container-data/certs/hze-nginx/server.key
```

- (Horizon Cloud Connector version 1.3 or earlier) Use the following command:

```
cp /root/server.key /etc/nginx/ssl/server.key
```

9 Verify the owner and permissions for the certificate and key file.

- (Horizon Cloud Connector version 1.4 or later) Use the following commands:

```
chown -R hze-nginx:hze-nginx /opt/container-data/certs/hze-nginx
chmod 644 /opt/container-data/certs/hze-nginx/server.crt
chmod 600 /opt/container-data/certs/hze-nginx/server.key
```
(Horizon Cloud Connector version 1.3 or earlier) Use the following commands:

```
chown -R root:root /etc/nginx/ssl
chmod -R 600 /etc/nginx/ssl
```

10 Verify that the issued FQDN in the certificate matches the server name directive in the server listen 443 block in the nginx configuration file.

- (Horizon Cloud Connector version 1.4 or later) The nginx configuration file is located at `/opt/container-data/conf/hze-nginx/nginx.conf`.
- (Horizon Cloud Connector version 1.3 or earlier) The nginx configuration file is located at `/etc/nginx/nginx.conf`.

11 Verify and restart nginx.

- (Horizon Cloud Connector version 1.4 or later) Use the following commands:

  `docker exec -i hze-nginx sudo nginx -t`
  `systemctl restart hze-nginx`

- (Horizon Cloud Connector version 1.3 or earlier) Use the following commands:

  `nginx -t`
  `systemctl restart nginx`

12 (Horizon Cloud Connector version 1.4 or later) Update the SSL thumbprints in the welcome screen.

  Use the following commands:

  `docker exec -i hze-core sudo /opt/vmware/bin/configure-welcome-screen.py`
  `/usr/bin/killall --quiet vami_login`

13 Test the new certificate by reloading the Horizon Cloud Connector user interface URL in a Web browser.

14 (Optional) If the certificate works correctly, remove the backed-up files.

- (Horizon Cloud Connector version 1.4 or later) Use the following commands:

  `rm /opt/container-data/certs/hze-nginx/server.crt.orig`
  `rm /opt/container-data/certs/hze-nginx/server.key.orig`
  `rm /opt/container-data/conf/hze-nginx/nginx.conf.orig`

- (Horizon Cloud Connector version 1.3 or earlier) use the following commands:

  `rm /etc/nginx/ssl/server.crt.orig`
  `rm /etc/nginx/ssl/server.key.orig`
  `rm /etc/nginx/nginx.conf.orig`
15 Remove the copied CA certificates and key files in the root directory.

Use the following commands:

```
rm /root/server.crt
rm /root/server.key
```

Synchronize the Horizon Cloud Connector Virtual Appliance with an NTP Server

To ensure that the Horizon Cloud Connector virtual appliance authenticates correctly with Horizon Cloud and the required Connection Server instances, you must synchronize the virtual appliance’s clock with a Network Time Protocol (NTP) server. Synchronize the clock on the Horizon Cloud Connector virtual appliance with the clock on the physical ESXi host on which the virtual appliance resides, after first ensuring that the host itself is properly synchronized with an NTP server.

Procedure

◆ (Preferred method) Synchronize the Horizon Cloud Connector virtual appliance with the physical ESXi host on which the virtual appliance resides.

a Verify that the ESXi host’s clock is properly synchronized with an NTP server.

For more information, see the VMware vSphere Documentation.

b Use vSphere Client to open the Edit Settings window of the Horizon Cloud Connector virtual appliance and enable the Synchronize Time with Host option.

For detailed instructions, see the VMware vSphere Documentation.

**Note** In Horizon Cloud Connector 1.5 and later, Synchronize Time with Host is enabled by default.

◆ (Alternative method) If you are unable to synchronize the Horizon Cloud Connector virtual appliance with the physical ESXi host, you can synchronize the virtual appliance directly with an NTP server.

**Note** The preferred method of time synchronization is to synchronize the virtual appliance with the physical ESXi host. Use the following steps only if you are unable to perform the preferred method.

a Open an SSH connection to the Horizon Cloud Connector virtual appliance, and log in as the root user.

b Using a text editor such as vi, open the timesyncd.conf file for editing.

```
vi /etc/systemd/timesyncd.conf
```
c Edit the [Time] section so that it resembles the following example. Replace *ntpAddress* with the domain name of the NTP server that you want to use.

```
[Time]
#FallbackNTP=time1.google.com time2.google.com time3.google.com time4.google.com
NTP=ntpAddress
```

Save your changes to the *timesyncd.conf* file and exit the text editor.

d Restart the virtual appliance's network service.

```
systemctl restart systemd-networkd
```

e Restart the virtual appliance's timesync service.

```
systemctl restart systemd-timesyncd
```

f Verify that the clock on the virtual appliance is now synchronized with the specified NTP server.

**Verify the Horizon Pod and Virtual Appliance Are Ready to Be Paired**

In this step in the workflow of onboarding a Horizon pod to Horizon Cloud, you run the *precheck.sh* diagnostic tool to validate that the pod and Horizon Cloud Connector are both ready for the pairing process. By first running the diagnostics and remediating any blocking issues found in system components and configurations, you can maximize your chances of success when pairing the pod with Horizon Cloud.

**Note** The *precheck.sh* diagnostic tool is only available for Horizon Cloud Connector 1.6 or later. If you have downloaded and deployed Horizon Cloud Connector 1.5 or earlier, disregard the following procedure and proceed directly to Complete Pairing the Horizon Pod with Horizon Cloud Using the Horizon Cloud Connector Configuration Portal.

The *precheck.sh* diagnostic tool validates the health of the services and components that are required to pair your Horizon pod successfully with Horizon Cloud. In addition, the tool checks whether:

- The configurations related to certificates and proxy settings are correct.
- Connectivity with Horizon Cloud and Connection Server can be established.
- Any SSL-related issues exist for Horizon Cloud Connector.

**Prerequisites**

Verify the following:

- You have completed the steps in Download and Deploy the Horizon Cloud Connector into the Pod's vSphere Environment, including the steps to Enable SSH Access to the Horizon Cloud Connector Prior to Pairing it with Your Horizon Connection Server.
- The Horizon Cloud Connector virtual appliance is powered on.
Procedure

1. Open an SSH session to your deployed Horizon Cloud Connector virtual appliance.

2. Run the diagnostic tool by using the following command. Replace `CS-FQDN` with the fully qualified domain name (FQDN) of the pod's Connection Server.

   ```
   /opt/vmware/bin/precheck.sh CS-FQDN
   ```

If the diagnostic tool discovers an issue that prevents the pairing of your Horizon pod with Horizon Cloud, it reports the following information:

- Name of the problem component or service
- Status of the problem component or service
- Associated error message and details
- Suggested remediation steps, if any, to restore the component or service to a healthy and ready state

**Note**  The diagnostic tool always reports one or both of the following expected conditions as part of its output. Both conditions are normal and expected at this stage of the onboarding workflow, and neither one blocks the pairing of the Horizon pod with Horizon Cloud.

- **Component/Service Name:** "Cloud Broker Client Service"
  - Status: "NOT_INITIALIZED"
  - Message: Service is not initialized.

  This condition pertains to the optional Universal Broker service, which remains in the NOT_INITIALIZED state until it is enabled as described in Setting Up Horizon Universal Broker for Multi-Cloud Assignments. You can still successfully pair the Horizon pod with Horizon Cloud when Universal Broker is in the NOT_INITIALIZED state. Therefore, this condition does not represent a blocking issue and you can disregard it.

- **Component/Service Name:** "Connector Client Service"
  - Status: "FAIL"
  - Message: Connector service is initialized post on-boarding.

  The Horizon Cloud Connector client service is initialized after you complete the pairing procedure, because the initialization process requires connectivity to Horizon Cloud. Therefore, the FAIL condition is expected at this stage of the onboarding workflow. After you pair the Horizon pod with Horizon Cloud, the Horizon Cloud Connector client service is initialized and the FAIL condition clears.
3 If the diagnostic tool reports an issue that interferes with the pairing process, investigate the affected component or service and perform the suggested remediation steps. As previously noted, you can disregard error conditions for "Cloud Broker Client Service" and "Connector Client Service" because they are not blocking issues.

As needed, repeat steps 2 and 3 to run the diagnostic tool again and troubleshoot problems until the tool does not report any blocking issues that prevent the pairing process. The Horizon pod and Horizon Cloud Connector are now ready for the pairing process.

**Note** If you attempt the pairing process without first clearing any blocking issues reported by the diagnostic tool, the pairing process might fail.

4 Continue with the pod onboarding workflow by following the steps in Complete Pairing the Horizon Pod with Horizon Cloud Using the Horizon Cloud Connector Configuration Portal.

**Complete Pairing the Horizon Pod with Horizon Cloud Using the Horizon Cloud Connector Configuration Portal**

In this step in the workflow of onboarding a Horizon pod to Horizon Cloud, you use the Horizon Cloud Connector configuration portal to specify details that the Horizon Cloud Connector uses to pair with the Horizon pod's Connection Server. Completing these steps successfully results in that pod connected to your Horizon Cloud tenant environment.

For an illustration of how the Horizon pod, Horizon Cloud, and Horizon Cloud Connector relate to each other in the pairing process, see the diagram shown in Connect Horizon Cloud Service with an Existing Horizon Pod to Use Horizon Subscription Licenses or Cloud-Hosted Services or Both.

**Note** If you encounter connectivity issues with the Horizon Cloud Connector configuration portal, see the troubleshooting information in VMware Knowledge Base (KB) article VMware Knowledge Base (KB) article 79859.

**Prerequisites**

Verify that you have completed the required preparation steps:

- Verify that you have completed the steps in Download and Deploy the Horizon Cloud Connector into the Pod's vSphere Environment.
- If you have deployed Horizon Cloud Connector 1.6 or later, verify that you have completed the steps in Verify the Horizon Pod and Virtual Appliance Are Ready to Be Paired.

Also, verify that:

- The Horizon Cloud Connector virtual appliance is powered on.
- You have the URL for displaying the browser-based Horizon Cloud Connector configuration portal. This URL is typically displayed in the appliance's blue console screen and is based on
the IP address of the virtual appliance, such as https://IP-address/ where IP-address is the appliance’s IP address. Alternatively, if you have mapped a fully qualified domain name (FQDN) to the Horizon Cloud Connector virtual appliance’s IP address in your DNS server, the configuration portal’s URL is that FQDN.

Verify that you have fulfilled all the items described in Preparing to Run the Onboarding Wizard to Pair a Horizon Pod with Horizon Cloud Using Horizon Cloud Connector, especially:

- You have met the DNS, Ports, and Protocols Requirements When Using Horizon Cloud Connector and a Horizon Pod when using the Horizon Cloud Connector to pair a Horizon pod with Horizon Cloud.

- Your DNS configuration in your network topology enables the deployed Horizon Cloud Connector to resolve the FQDN of the pod’s Connection Server. If the deployed Horizon Cloud Connector cannot resolve the Connection Server using DNS, the onboarding wizard will encounter an unexpected error at the step where you enter the pod’s domain credentials.

- The Horizon Cloud Connector virtual appliance must reach the Internet to talk to the Horizon Cloud control plane and display the browser-based configuration portal. If your environment requires use of a proxy server and proxy configuration with deployed appliances, verify that you have configured the deployed Horizon Cloud Connector appliance with the proxy settings required for your environment. See the proxy-related information in Preparing to Run the Onboarding Wizard to Pair a Horizon Pod with Horizon Cloud Using Horizon Cloud Connector, Horizon Cloud Connector Known Considerations, and Modifying Proxy Settings for Horizon Cloud Connector 1.5 or Earlier.

- You have the credentials of the My VMware account that is associated with the Horizon Cloud customer account to which you are pairing the pod. As described in Connect Horizon Cloud Service with an Existing Horizon Pod to Use Horizon Subscription Licenses or Cloud-Hosted Services or Both, a My VMware account is required to authenticate with the cloud management plane to set up the connector and establish the connection to use that license for the Horizon subscription offerings.

Procedure

1. Obtain the URL for launching the web-based configuration portal from the appliance’s blue console screen.
2 Using a browser, navigate to the URL that you obtained from the appliance's blue console screen.

**Important** In this step, the Horizon Cloud Connector makes a connection to Horizon Cloud to display the login screen, which will be used to authenticate your My VMware account credentials with the cloud control plane. This connection is outbound HTTPS using port 443. If you do not see a login screen, verify that you have met the DNS, Ports, and Protocols Requirements When Using Horizon Cloud Connector and a Horizon Pod.

The login screen for logging in to the Horizon Cloud Connector configuration portal appears.

3 In the login screen, enter the My VMware account credentials and click **Login**.

The following screenshot is an example of the login screen with the credentials entered before clicking **Login**.

![Login Screen Example](image)

When the Terms of Service message appears, click **Accept** to continue.

The configuration portal displays the first step of the pod onboarding wizard. The following screenshot illustrates this step before any fields are filled in.
In the **Connect to Horizon Connection Server** field, enter the FQDN of the pod's Connection Server instance to which you are pairing the Horizon Cloud Connector.

As you type in the field, the **Connect** button appears.

When you have typed in the FQDN, click **Connect**.

The Horizon Cloud Connector attempts to communicate with the specified Connection Server and retrieve its certificate information. This process can take a few minutes. When communication is established, the page displays the retrieved certificate information.

If the Connection Server does not have a valid Root CA certificate, a warning message appears saying that the certificate cannot be automatically validated and you must confirm its validity by clicking the check box. The following screenshot is an example of this situation.
If you see this message, verify that the displayed certificate information is accurate and click the check box so you can proceed to the next step.

**Note** If the Connection Server has a valid Root CA certificate, the wizard automatically validates the information and you can proceed to the next step.

The following screenshot illustrates the screen after clicking the check box.

6 In the credentials section, type the domain name and the administrator credentials used by the Connection Server and click **Connect**.

This administrator account must have the Horizon predefined Administrators role with root access for the pod. For more information about the requirements of the administrator account, see [Preparing to Run the Onboarding Wizard to Pair a Horizon Pod with Horizon Cloud Using Horizon Cloud Connector](#).

The following screenshot illustrates this area of the screen.
Note  At this point, the system detects whether the specified Connection Server instance is already paired with another instance of the Horizon Cloud Connector. In this case, the page displays a message asking whether you want to delete the existing pairing and pair this Connection Server with the new Horizon Cloud Connector instance. If the new Horizon Cloud Connector instance is at a later version than the existing instance, you also have the option to perform an appliance update by copying the existing configuration into the new Horizon Cloud Connector instance, as described in Manually Update the Horizon Cloud Connector Virtual Appliance.

Click the appropriate action button in the message to continue pairing the pod using the following steps.

Step 2 of the wizard appears.

7 In this wizard step, provide details about the pod.

The following screenshot is an example of this step filled in.
These details are used in the cloud management plane to associate the paired Connection Server instance and Horizon Cloud Connector with your Horizon Cloud tenant environment. As an example, the specified name, location, and description will be visible in the administrative console so that you can identify this pod from your other pods that are connected to the control plane.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Type a friendly name to identify this pod in your Horizon Cloud tenant environment.</td>
</tr>
<tr>
<td>Data Center Location</td>
<td>Select an existing location or click New to specify a new one to use for this pod. In the cloud-based administrative console, your pods are grouped and displayed according to the locations you specify. In the City Name text box, start typing the name of a city. The system automatically displays world city names in its backend geography lookup table that match your entered characters, and you can choose a city from that list. <strong>Note</strong> You must select a city from the system's autocomplete list.</td>
</tr>
<tr>
<td>Description</td>
<td>Optional: Enter a description for this pod.</td>
</tr>
<tr>
<td>Pod deployed in VMware Cloud on AWS</td>
<td>Select this check box if the pod is deployed in a VMware Cloud on AWS software-defined data center (SDDC).</td>
</tr>
</tbody>
</table>

8 Proceed to the next wizard step by clicking **Save**.

The wizard's configuring step appears. The system checks the connection to the specified Connection Server instance and completes the final configuration steps. The following screenshot is an example of this step.

![Setup 3 of 3: Configuring](image)

When the system determines the pod is successfully connected to the Horizon Cloud control plane, a congratulations screen appears with some guidance text and action buttons for post-configuration management tasks. The following screenshot is an example of this screen.
The action buttons on the congratulations screen can vary depending on the version of Horizon Cloud Connector that you have deployed:

- The **Configure vCenter Server and Network Details** button is available only if you have deployed Horizon Cloud Connector 1.7 or later.

- The **Configure Automatic Cloud Connector Updates** button is available only if you have deployed Horizon Cloud Connector 1.6 or earlier and your Horizon Cloud tenant account is configured for automated updates of the Horizon Cloud Connector. For details about that feature, see the topic **Automated Updates of the Horizon Cloud Connector Virtual Appliance** in the **Administration Guide**.
**Important** If you have deployed Horizon Cloud Connector 1.7 or later, the Horizon Cloud Connector vCenter Server Details window automatically opens in front of the congratulations screen. To support automated updates and the proper functionality of Horizon Cloud Connector, you must enter these mandatory vCenter Server and network details as described in the topic Automated Updates of the Horizon Cloud Connector Virtual Appliance in the Administration Guide. If you close the Horizon Cloud Connector vCenter Server Details window without entering the mandatory details, you will encounter repeated warning messages until you click the Configure vCenter Server and Network Details button and enter the details.
Results

When you reach this point, the pairing workflow is complete. At this point, VMware activates the subscription license, typically within 30 minutes after you paired the pod with the cloud control plane. When VMware has activated the subscription license, a message appears in your pod's web-based management console that indicates your pod is using the subscription type of license. The following screenshot is a sample illustration.

Attention  If 48 hours pass and you still do not see the Connected to License Service message in your pod’s web-based management console's licensing area, contact your VMware representative.

What to do next

From this point, the pod is successfully paired with Horizon Cloud. For details about those Horizon Cloud Connector administrative and maintenance tasks that are typically done from this point on, see the topic Typical Administrative and Maintenance Tasks You Perform on the Horizon Cloud Connector After the Horizon Pod is Paired with Horizon Cloud in the Administration Guide.

When You Choose Microsoft Azure Cloud Capacity for Your Very First Pod Deployment

You connect Horizon Cloud to your Microsoft Azure subscription to manage and deliver Microsoft Windows 10 VDI desktops and virtual RDSH Windows VMs for session-based desktops and remote applications. Setting up the environment involves deploying the required VMware software into your Microsoft Azure capacity. The deployed VMware software creates an appropriately configured entity, called a pod, which pairs with the control plane. After the pod is
deployed, then you use the control plane to provision VDI desktops and RDSHs, and entitle access to desktops and remote applications to your end users.

**Pod Deployment in Microsoft Azure**

The pod deployed by Horizon Cloud into Microsoft Azure has a physical regional location in a Microsoft Azure cloud. In the pod deployment wizard, you select where to place the pod, according to the regions available for your particular Microsoft Azure subscription. You also select an existing virtual network (VNet) that the pod will use in your selected region. You have the option to deploy an external gateway configuration with the pod, with that external gateway's resources deployed either into the same VNet as the pod or into a separate VNet that is peered with the pod's VNet.

**Note**  You preconfigure your Microsoft Azure environment with the pod VNet (and with the external gateway VNet if using that configuration option). You can either create in advance those subnets that the pod and external gateway configuration require, or let the pod deployer create the subnets during deployment. If you do not create the subnets in advance, the pod deployer creates the subnets as it deploys the required VMs and resources into your environment. If you choose to have the pod deployer create its required subnets, you have to know what IP address spaces you want to use for the pod's subnets before you start the deployment wizard. If you choose to create the subnets in advance, you must ensure they meet certain requirements before you start the deployment process. For details about requirements when you create the subnets in advance, see In Advance of Pod Deployment, Create the Horizon Cloud Pod's Required Subnets on your VNet in Microsoft Azure and When Using Existing Subnets for a Horizon Cloud Pod in Microsoft Azure.
You can deploy more than one pod into Microsoft Azure and manage all of them from the same administrative console. The pods you deploy after the first one can reuse the same VNet as your first pod or use different VNets. Also, each pod can be in a different Microsoft Azure region, using a VNet in each region.

**Important** This pod in Microsoft Azure is not a tenant. This pod does not adhere to the exact same set of characteristics that defines a tenant and which you would expect from a tenant. For example, even though a tenant would have a one-to-one mapping to an Active Directory domain and be isolated from other tenants, all of the Horizon Cloud pods in Microsoft Azure that are deployed using the same Horizon Cloud customer account record need to be able to reach the same Active Directory servers and the DNS configuration needs to resolve all of those Active Directory domains.

To do multi-tenancy, you would set up multiple Horizon Cloud customer account records. The Horizon Cloud customer account record, which is created when you registered with VMware to use the Horizon Cloud Service and is associated with your My VMware credentials, is more like a tenant. A Horizon Cloud customer account record is isolated from other Horizon Cloud customer account records. A single customer account record maps to multiple pods, and when someone uses any of the account credentials associated with that customer account record to log in to the administrative console, the console reflects all of the pods that are mapped to that customer account record.

The pod deployment process automatically creates a set of resource groups in your Microsoft Azure capacity. Resource groups are used to organize the assets that the environment needs and creates, such as:

- VMs for the pod’s manager instance (multiple VMs for a pod that is enabled for high availability)
- VMs for the Unified Access Gateway instances and their load balancers
- VM for the connector VM in the external gateway configuration when you deploy that configuration in a VNet separate from the pod’s VNet
- VMs for the RDSH-capable golden images
- VMs for the VDI desktop golden images
- VMs for the assignable (published, sealed) images that are made from the golden images
- VMs for the RDSH farms that provide the RDSH desktops and remote applications
- VMs for the VDI desktops
- Additional assets that the VMs and the environment require for supported operations, such as network interfaces, IP addresses, disks, key vaults, Microsoft Azure Database for PostgreSQL server resource, and various items along those lines. The pod deployment process can create the required virtual subnets also, using the values you specify in the deployment wizard.
The following diagram illustrates a deployed pod that is enabled for high availability, has both the external and internal types of gateway configurations, and where the external gateway resides in the same VNet as the pod itself. In this diagram, RG means resource group. The Unified Access Gateway instances in the external gateway configuration have NICs on the demilitarized (DMZ) network. With an external gateway configuration, you can have your end users located in the Internet, outside your corporate network, accessing their pod-provisioned virtual desktops and applications through that configuration. With an internal gateway configuration, you can have your end users located in your intranet, inside your corporate network, making trusted connections to their pod-provisioned virtual desktops and applications through that gateway. The pod deployment wizard provides the option to deploy the pod with both configurations up front. Alternatively, you can deploy the pod with only one gateway configuration or with none at all, and edit the deployed pod to add the non-chosen gateway configuration later.

You can also choose not to enable the high availability option in the deployment wizard, and then edit the deployed pod later to enable high availability on it. Starting in this release, a new pod is always deployed with a Microsoft Azure Database for PostgreSQL server resource and a pod load balancer, even when you do not enable the high availability option in the wizard. Having those resources available allows for enabling high availability on an already deployed pod. The second pod manager VM is only deployed when high availability is enabled on the pod. For more information, see the High Availability and Your Horizon Cloud Pod in Microsoft Azure documentation topic in the Administration Guide.
Figure 2-1. Illustration of the Horizon Cloud Pod Architecture for a Pod with High Availability Enabled and Configured with Both External and Internal Gateway Configurations, the External Gateway Deployed into the Same VNet as the Pod, Three NICs on the External Gateway VMs, Two NICs on the Internal Gateway VMs, and a Public IP Enabled for the External Gateway’s Load Balancer
The following diagram illustrates the resources that are deployed when you choose the option to have the external gateway residing in its own VNet, separate from the pod's VNet. The two VNETs must be peered. This diagram also applies when you choose the option to have the external gateway's resources deployed using a Microsoft Azure subscription that is different than the one used for the pod. Because VNETs cannot cross subscriptions, choosing to deploy the external gateway into its own subscription is a subset of choosing the external gateway to reside in its own VNet.

**Tip** Deploying the external gateway configuration into its own VNet gives you the ability to deploy these Horizon Cloud pods into complex Microsoft Azure environments that use hub-spoke network topology in Microsoft Azure.

**Figure 2-2. Illustration of the External Gateway's Architecture Elements When the External Gateway is Deployed into Its Own VNet, Separate from the Pod's VNet**

**Microsoft Azure Terminology and References**

The VMware Horizon Cloud Service on Microsoft Azure product documentation uses the applicable Microsoft Azure terminology as appropriate in the descriptions and task steps of the VMware Horizon Cloud Service on Microsoft Azure workflows. If the Microsoft Azure terminology is unfamiliar to you, you can use the following applicable references in the Microsoft Azure product documentation to learn more.

**Note** All capitalization and spelling in the citations below follow the same capitalization and spelling found in the linked-to articles in the Microsoft Azure documentation itself.
<table>
<thead>
<tr>
<th>Useful Microsoft Azure References</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Microsoft Azure glossary: A dictionary of cloud terminology on the Azure platform</td>
<td>Use this glossary to learn the meaning of terms as used in the Microsoft Azure cloud context, for terms such as load balancer, region, resource group, subscription, virtual machine, and virtual network (vnet). <strong>Note</strong> The Microsoft Azure glossary does not include the term service principal because the service principal is a resource automatically created in Microsoft Azure when an application registration is created in Microsoft Azure. The reason why you create an application registration in your Microsoft Azure subscription is because that is the way you authorize Horizon Cloud as an application to use your Microsoft Azure capacity. The application registration and its companion service principal enable the Horizon Cloud cloud service acting as an application to access resources in your Microsoft Azure subscription. Use the next reference below to learn about applications and service principals that can access resources in Microsoft Azure.</td>
</tr>
<tr>
<td>Use portal to create an Azure Active Directory application and service principal that can access resources</td>
<td>Use this article to learn about the relationship between an application and a service principal in a Microsoft Azure cloud.</td>
</tr>
<tr>
<td>Azure Resource Manager overview</td>
<td>Use this article to learn about the relationships between resources, resource groups, and the Resource Manager in Microsoft Azure.</td>
</tr>
<tr>
<td>Azure VNet</td>
<td>Use this article to learn about the Azure Virtual Network (VNet) service in Microsoft Azure. See also Azure Virtual Network FAQs.</td>
</tr>
<tr>
<td>Azure VNet Peering</td>
<td>Use this article to learn about virtual network peering in Microsoft Azure.</td>
</tr>
<tr>
<td>Hub-spoke network topology in Azure</td>
<td>Use this article to learn about hub-spoke network topology in Microsoft Azure.</td>
</tr>
<tr>
<td>Microsoft Azure ExpressRoute Overview</td>
<td>Use this article to learn about Microsoft Azure ExpressRoute and how you can use it to establish connections between your on-premises networks, Microsoft Azure, and your Horizon Cloud pods.</td>
</tr>
<tr>
<td>About VPN Gateway Planning and design for VPN Gateway Create a Site-To-Site connection in the Azure portal</td>
<td>Use these articles to learn about how to configure VPNs in Microsoft Azure.</td>
</tr>
<tr>
<td>What is Azure Load Balancer?</td>
<td>Use this article to learn about the Azure load balancers that are deployed for a pod: the load balancer for the pod manager VMs and the load balancers for the gateway configurations.</td>
</tr>
<tr>
<td>What is Azure Database for PostgreSQL?</td>
<td>Use this article to learn about the Microsoft Azure Database for PostgreSQL service.</td>
</tr>
<tr>
<td>What is Windows Virtual Desktop?</td>
<td>Use this article to learn about Microsoft Windows Virtual Desktop and how it relates to Microsoft Windows 10 Enterprise multi-session and Microsoft Windows 7 Enterprise with Extended Security Updates. When your Horizon Cloud tenant account has the configuration for Horizon Cloud Service on Microsoft Azure extending Microsoft Windows Virtual Desktop, support is provided for using Microsoft Windows 10 Enterprise multi-session and Microsoft Windows 7 Enterprise with your pods deployed in Microsoft Azure.</td>
</tr>
</tbody>
</table>
Additional VMware Resources

The following resources provide in-depth technical details about the service.

<table>
<thead>
<tr>
<th>Additional VMware Technical Resources</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Requirements Checklist</td>
<td>Use this checklist to learn about the assets you need to obtain and configure prior to beginning the pod deployment process.</td>
</tr>
<tr>
<td>Networking and Active Directory Considerations on Microsoft Azure with VMware Horizon Cloud</td>
<td>Use this article to learn about the various options and best practices for networking connections and using Microsoft Active Directory with your Horizon Cloud pods in Microsoft Azure.</td>
</tr>
<tr>
<td>Horizon Cloud Service on Microsoft Azure Security Considerations</td>
<td>Use this article to obtain information about the security details of the environment and the types of data stored.</td>
</tr>
<tr>
<td>Horizon Cloud on Microsoft Azure: RDS Desktop and App Scalability (technical paper download)</td>
<td>Use this article to gain insight from analyses on RDS desktop and remote application scalability and optimal user densities, as well as cost considerations related to farm deployment and power management settings.</td>
</tr>
</tbody>
</table>

High-Level Workflow for When Your Very First Horizon Cloud Cloud-Connected Pod is from Using the Pod Deployer to Deploy a Pod into Microsoft Azure

This is a high-level list of the steps when you are arriving at your very first cloud-connected pod by running the pod deployer to deploy a Horizon Cloud pod into your Microsoft Azure capacity. After that first cloud-connected pod is fully deployed and you have completed the steps to register Horizon Cloud with the pod’s intended Active Director domain, you can use all the features provided Horizon Cloud, especially for provisioning VDI desktops, RDSH session-based desktops, or RDSH-based remote applications to your end users from that pod. When your customer account is configured to use App Volumes features with your pods in Microsoft Azure, you can also provision applications from those App Volumes features and entitle those to your end users.

Perform the following steps when you are deploying your very first cloud-connected pod and you are using the pod deployment wizard to deploy it into Microsoft Azure.

1. Fulfill the prerequisites. See VMware Horizon Cloud Service on Microsoft Azure Requirements Checklist For New Pod Deployments - Updated As Appropriate for Pods That Deploy Starting From the July 2020 Service Release.

2. Perform the preparatory tasks outside of Horizon Cloud. See Preparing to Deploy a Horizon Cloud Pod Into Microsoft Azure.

3. Verify you meet the DNS, ports, and protocol requirements for deploying the pod. See DNS Requirements for a Horizon Cloud Pod in Microsoft Azure and Ports and Protocols Requirements for a Horizon Cloud Pod at the September 2019 Release’s Manifest or Later.

4. Deploy the pod. See Deploy a Horizon Cloud Pod into Microsoft Azure.
5 Register your Active Directory domain with the deployed pod, which includes providing the name of a domain-join account. See Performing Your First Active Directory Domain Registration in the Administration Guide.

6 Give the Horizon Cloud Super Administrators role to an Active Directory group that includes that domain-join account as a member.

**Important** You must ensure that the domain join account you enter when registering the domain is also in one of the Active Directory groups to which you assign the Horizon Cloud Super Administrators role. The system's domain-join operations with the pod depend on the domain join account having the Horizon Cloud Super Administrators role. See Assign Administrative Roles to Active Directory Groups.

7 Select the type of brokering you want your tenant's pods to use when brokering pod-provisioned resources to your end users. See the topic Introduction to Universal Broker and Single-Pod Brokering and its related topics and subtopics in the Administration Guide.

**Attention** Completing this brokering selection step before deploying additional pods into Microsoft Azure is a best practice.

8 If you plan to use Workspace ONE Access with the pod or you plan to have Horizon Clients connecting directly to the pod (not through a pod gateway configuration), then perform these steps:

- In your DNS server, map a fully qualified domain name (FQDN) to the pod manager's Microsoft Azure load balancer IP address
- Obtain an SSL certificate based on that mapped FQDN
You will upload an SSL certificate to the pod that is based on the FQDN that you’ve mapped to the pod manager’s Microsoft Azure load balancer IP address in your DNS so that connections that go to the pod manager VMs will make trusted connections. Such connections include Horizon Clients, for your users that you give that mapped FQDN to, and the Workspace ONE Access Connector that is used when you integrate Workspace ONE Access with the pod. The Workspace ONE Access Connector must connect to the pod using an FQDN that is mapped to the pod manager’s Microsoft Azure load balancer IP address.

**Attention** When you are integrating Workspace ONE Access with the pod, you must upload an SSL certificate to the pod and configure your Workspace ONE Access to point to the pod, not to the pod’s Unified Gateway Access configurations.

However, bear in mind that when you have uploaded an SSL certificate based on your DNS-mapped FQDN, if you try to connect by directly typing that FQDN into a browser — not going through a properly configured Workspace ONE Access — that pure FQDN use will appear as untrusted connections to the browser. The reason is because simply loading that FQDN into a browser is a connection using HTML Access (Blast), and that is how HTML Access (Blast) behaves. As a result, when you load that FQDN into a browser, it displays the typical untrusted certificate error.

In the absence of having Workspace ONE Access, to have connections using HTML Access (Blast) — using a browser basically — avoid the displayed untrusted certificate error, you must put a gateway configuration on the pod and have those connections use the load balancer and Unified Access Gateway instances from that gateway configuration. If you do not want to expose your FQDN to the Internet, you can deploy an internal Unified Access Gateway configuration. This internal Unified Access Gateway configuration uses a Microsoft Azure internal load balancer to which end users who are internal to your corporate network can make their connections.

9 Upload an SSL certificate to the pod directly, using the pod’s summary page in the administrative console, if you plan to have one or both of the use cases described in the preceding step. See Configure SSL Certificates on the Horizon Cloud Pod’s Manager VMs.

**Tip** If the only access use case you will ever want to support is where connections will go to the pod’s Unified Access Gateway instances through the load balancer connected to those instances, then uploading the SSL certificate to the pod directly is superfluous. Still, performing the immediately preceding step above and this step here is a recommended practice, because it ensures that if you do one day give out that FQDN to users to enter in their Horizon Clients, those clients can have trusted connections. Performing the immediately preceding step and this one here also provides you the ability to one day more quickly integrate the pod with Workspace ONE Access because you would have the FQDN mapped and the SSL certificate already in place on the pod.
10 Import a base image. On the Imported VMs page, use the **Reset Agent Pairing** action to pair the new image with Horizon Cloud. See *Creating Desktop Images for a Horizon Cloud Pod in Microsoft Azure*.

**Note**  Tech Preview: Use of Microsoft Windows 10 Enterprise multi-session operating system with App Volumes is currently in tech preview. To import a base image for that use case, use these steps instead: *Tech Preview - How to Configure a Windows 10 Multi-Session VM for Use with App Volumes Features in this Horizon Cloud Release*.

11 Depending on the type of end-user assignment the image will ultimately be used for, perform one or more of the following steps as appropriate.

- In an image VM that will be used for provisioning VDI desktops or native applications, install the third-party applications you want your end users to use in their VDI desktops, and configure other applicable customizations, such as setting desktop wallpaper, installing the NVIDIA GPU drivers (for GPU-enabled images), and so on. Also optimize the image for Microsoft Sysprep best practices, if not done as part of the import image process. See *Customize the Image VM’s Windows Operating System* and its subtopics, and *Install NVIDIA Graphics Drivers in a GPU-Enabled Image* in the *Administration Guide*.

  **Tip**  To further tune the image VM to provide an improved configuration for using VMware Blast Extreme in VDI use cases, a best practice is to read the *VMware Blast Extreme Optimization Guide* and perform additional tuning for codec options in the image according to that guide’s recommendations for codec options.

- In a RDS-capable image that will be used for provisioning RDSH-based session desktops and remote applications, install the third-party applications you want to provide to your end users from that RDS image and configure other applicable customizations, such as setting desktop wallpaper, installing the NVIDIA GPU drivers (for GPU-enabled images), and so on. Also optimize the image for Microsoft Sysprep best practices, if not done as part of the import image process. If the imported VM is running one of the Microsoft Windows 10 Enterprise multi-session systems that includes Office 365 ProPlus by default, you should verify the VM is configured for shared computer activation for Office 365 ProPlus as described in the Microsoft documentation topic *Overview of shared computer activation for Office 365 ProPlus*. If Office 365 ProPlus is not configured for shared computer activation in the imported VM, use the method described in that Microsoft document that is appropriate for your situation. See *Customize the Image VM’s Windows Operating System* and its subtopics, and *Install NVIDIA Graphics Drivers in a GPU-Enabled Image* in the *Administration Guide*.

  **Tip**  To further tune the image VM to provide an improved configuration for using VMware Blast Extreme in VDI use cases, a best practice is to read the *VMware Blast Extreme Optimization Guide* and perform additional tuning for codec options in the image according to that guide's recommendations for codec options.
12 Convert that image into an assignable image, also known as sealing or publishing the image. See Convert a Configured Virtual Machine to an Assignable Image in the Administration Guide.

13 To provision session-based RDSH desktops and remote applications from a published server image:
   a Create a desktops RDSH farm to provide session desktops, and then create assignments to entitle end users to use those desktops. See Create a Farm and Create an RDSH Session Desktop Assignment in the Administration Guide.
   b Create an applications RDSH farm to provide remote applications, add the applications to your application inventory, and then create assignments to entitle end users to use those remote applications. See Create a Farm, Import New Remote Applications from RDSH Farms, and Create a Remote Application Assignment in the Administration Guide.

14 To provision VDI desktops from a published VDI desktop image, create a dedicated or floating VDI desktop assignment. See About Desktop Assignments for Your Horizon Cloud Environment's Pods in Microsoft Azure and its VDI-related subtopics in the Administration Guide.

15 To provision App Volumes applications to your end users, add the App Volumes applications to your application inventory and create an application assignment to entitle end users to use those applications. Then create a desktop assignment based on the same published image to entitle those end users to desktops in which they can launch those applications. See App Volumes Applications - Overview and Prerequisites in the Administration Guide.

16 When a pod is deployed with a gateway configuration, you must create a CNAME record in your DNS server that maps the fully qualified domain name (FQDN) that you entered in the deployment wizard to the appropriate Microsoft Azure load balancer resource that is configured in the pod for that gateway.

   ■ For an external gateway enabled with a public IP address, map the FQDN that you entered in the deployment wizard to the gateway's Microsoft Azure load balancer resource's auto-generated public FQDN. Your DNS server record maps that Microsoft Azure load balancer's auto-generated public FQDN with the FQDN that your end users will use, and which is used in the uploaded certificate. The following code line demonstrates an example. You locate the ID to use from the pod's details page in the console, after you have registered the Active Directory domain. If the external gateway was deployed in its own VNet, use the ID that is displayed in the Deployment ID field.

   ourApps.ourOrg.example.com   vwm-hcs-ID-uag.region.cloudapp.azure.com
For an internal gateway or an external gateway without a public IP address, map the FQDN that you entered in the deployment wizard to the gateway's Microsoft Azure load balancer resource's private IP address. Your DNS server record maps that Microsoft Azure load balancer's IP address with the FQDN that your end users will use, and which is used in the uploaded certificate. The following code line demonstrates an example.

```
ourApps.ourOrg.example.com   Azure-load-balancer-private-IP
```

After the pod is onboarded and you can access the Capacity page in the administrative console, navigate to the Capacity page to see the `vmw-hcs-ID-uag.region.cloudapp.azure.com` value needed to map your FQDN in your DNS.

For details on how to locate the Microsoft Azure load balancer's FQDN in the console, see Obtain the Pod Gateway's Load Balancer Information to Map in Your DNS Server in the Administration Guide.

17 When a pod is deployed to have RADIUS two-factor authentication for the pod's gateways, you must complete the following tasks:

- If you configured an external gateway with RADIUS settings and that RADIUS server is not reachable within the same VNet as used by the pod, or within the peered VNet topology if you deployed the external gateway into its own VNet, verify, configure that RADIUS server to allow client connections from the IP address of the external gateway's load balancer. In an external gateway configuration, the Unified Access Gateway instances attempt contact with the RADIUS server using that load balancer address. To allow the connections, ensure the load balancer resource's IP address that is in that external gateway's resource group is specified as a client in your RADIUS server configuration.

- If you configured an internal gateway, or an external gateway and your RADIUS server is reachable within the same VNet as used by the pod, configure the RADIUS server to allow connections from the appropriate NICs that were created in the gateway's resource group in Microsoft Azure that must communicate with the RADIUS server. Your network administrator determines the RADIUS server's network visibility to the pod's Azure Virtual Network and subnets. Your RADIUS server must allow client connections from the IP addresses of those gateway NICs that correspond to the subnet for which your network administrator has given network visibility to the RADIUS server. The gateway's resource group in Microsoft Azure has four NICs that correspond to that subnet, two that are currently active for the two Unified Access Gateway instances and two that are idle and will become the active ones after the pod goes through an update. To support connectivity between the gateway and the RADIUS server both for ongoing pod operations and after each pod update, ensure the IP addresses of those four NICs are specified as clients in the RADIUS server configuration.

For information on how to obtain those IP addresses, see Update Your RADIUS System with the Required Horizon Cloud Pod Gateway Information.
After the above workflow steps are completed, your end users can launch their entitled desktops and remote applications using your FQDN in the Horizon Client or with HTML Access.

You can find in-depth details on how to accomplish each workflow step in the topics that are linked from each step above or in the companion guide. See the topic Administration of Your Horizon Cloud Tenant Environment and Your Fleet of Onboarded Pods and subtopics.

Preparing to Deploy a Horizon Cloud Pod Into Microsoft Azure

Before you log in to the Horizon Cloud Administration Console and run the pod deployment wizard for the first time, you must perform these preparatory tasks.

1. Fulfill the prerequisites described in the VMware Horizon Cloud Service on Microsoft Azure Requirements Checklist For New Pod Deployments - Updated As Appropriate for Pods That Deploy Starting From the July 2020 Service Release, especially:

   - Ensure your Microsoft Azure account and subscription encompasses the pod's required number and sizes of virtual machines, including the optional Unified Access Gateway configurations if you plan to deploy those. See Microsoft Azure Virtual Machine Requirements for a Horizon Cloud Pod in Microsoft Azure.

   If you plan to deploy the pod with an external gateway configuration that uses its own subscription, separate from the pod's subscription, ensure that other subscription encompasses the external gateway's required number and sizes of virtual machines. For this use case, that separate subscription will need its own VNet, because VNets do not span subscriptions. Also, this subscription must be in the same region as the pod's subscription because the supported VNet topology is connecting VNets within the same Microsoft Azure region.

   - As described in that the VMware Horizon Cloud Service on Microsoft Azure Requirements Checklist For New Pod Deployments - Updated As Appropriate for Pods That Deploy Starting From the July 2020 Service Release, ensure your subscription does not restrict use of Azure StorageV1 account type, does not restrict creation of untagged resource groups or require specific tags on its resource groups, and has no Azure Policies on it that would block, deny, or restrict creation of the pod's components in that subscription.

   Caution  The pod deployment process will fail early on if your subscription restricts those items, because the first step of creating the temporary jump box's resource group and deploying the jump box will fail to complete. Therefore, if your pod deployment process times out after two hours, first check if your subscription has Azure Policies in place that would block, deny, or restrict creation of resource groups based on particular criteria.

   - Ensure a virtual network (VNet) exists in the region in which you are going to deploy the pod and that virtual network meets the requirements for a Horizon Cloud pod. If you do not have an existing VNet, create one that meets the requirements. See Configure the Required Virtual Network in Microsoft Azure.
If you plan to deploy the pod with an external gateway configuration that uses its own VNet, separate from the pod's VNet — or that uses its own subscription separate from the pod's subscription, ensure that VNet exists in the same region as the pod's VNet, and that it meets the Configure the Required Virtual Network in Microsoft Azure. For this use case, those two VNets must be peered.

**Important** Not all Microsoft Azure regions support GPU-enabled virtual machines. If you want to use the pod for GPU-capable desktops or remote applications, ensure that the Microsoft Azure region you select for the pod provides for those NV-series VM types that you want to use and which are supported in this Horizon Cloud release. See the Microsoft documentation at https://azure.microsoft.com/en-us/regions/services/ for details.

- If you want to manually create the subnets for the pod on your VNet in advance of deploying the pod, ensure that the required number of subnets is created on the VNet, that their address spaces meet the Configure the Required Virtual Network in Microsoft Azure, and that they are empty of resources. In Advance of Pod Deployment, Create the Horizon Cloud Pod's Required Subnets on your VNet in Microsoft Azure.

**Caution** These subnets you create on your VNet for a pod deployment must be empty. You can create the subnets prior to deploying the pod, but do not put any resources on those subnets or otherwise use any of the IP addresses. If an IP address is already in use on the subnets, the pod might fail to deploy.

If you do not want to create the subnets in advance, the pod deployment process will create them using the CIDR information you enter into the on-screen wizard.

- Ensure that virtual network is configured to point to a valid Domain Name Services (DNS) server that is resolving external names. See Configure the DNS Server Settings Needed by the VNet Topology You Will Use for Your Horizon Cloud Pods in Microsoft Azure.

**Important** The pod deployment process requires external and internal name resolution. If the VNet points to a DNS server that cannot resolve external names, the deployment process will fail.

- If you plan to deploy the pod with an external gateway configuration into an existing resource group that you create in a subscription separate from the pod's subscription — instead of having the deployer auto-create that resource group — ensure that resource group exists in that subscription before you start the pod deployment wizard. Decide whether to set the permissions that Horizon Cloud needs at the resource-group level or at the subscription level. See Operations Required by Horizon Cloud in Your Microsoft Azure Subscriptions.

- Ensure you have an Active Directory setup that is supported for use with this release, your virtual network can reach it, and the DNS server can resolve its name. See Active Directory Domain Configurations.
2 Create the required number of service principals, according to your planned deployment options. If you are deploying the pod’s external gateway configuration into its own subscription, then you need a service principal for that subscription as well as for the subscription used for the pod itself. For detailed steps, see Create the Required Service Principal Needed by the Horizon Cloud Pod Deployer by Creating an Application Registration.

**Important** Each service principal that you configure for Horizon Cloud’s use must be assigned an appropriate role in that service principal’s associated subscription. The role to a service principal must allow the actions that Horizon Cloud needs to operated on the Horizon Cloud managed resources in that service principal’s associated Microsoft Azure subscription. The service principal for the pod’s subscription needs a role that allows for actions to successfully deploy the pod, to operate on the pod and the pod-managed resources to fulfill the administrator workflows initiated using the administrative console, and to maintain the pod over time. When using a separate subscription for the pod’s external Unified Access Gateway configuration, the service principal for that subscription needs a role that allows for actions to successfully deploy the resources needed for that gateway configuration, to operate on those Horizon Cloud-managed resources to fulfill the administrator workflows, and to maintain those gateway-related resources over time.

As described in Operations Required by Horizon Cloud in Your Microsoft Azure Subscriptions, the service principal must be granted access using one of the following methods:

- At the subscription level, assign the Contributor role. The Contributor role is one of the Microsoft Azure built-in roles. The Contributor role is described in Built-in roles for Azure resources in the Microsoft Azure documentation.

- At the subscription level, assign a custom role that you have set up to provide the service principal with Operations Required by Horizon Cloud in Your Microsoft Azure Subscriptions that Horizon Cloud needs for deployment of the pod-related resources and for ongoing administrator-initiated workflows and pod maintenance operations.

- When using a separate subscription for the external Unified Access Gateway configuration and deploying into an existing resource group, an valid combination is to grant access to the service principal to access that resource group and associated VNet using a role that provides narrow-scope permissions plus grant access for the service principal to access the subscription using the built-in Reader role.

3 From the Microsoft Azure portal, for the pod’s subscription and the subscription for its external gateway (if using that deployment option), get the values for the Microsoft Azure subscription ID, application ID, application authentication key, and Microsoft Azure AD Directory ID from the Microsoft Azure portal. These resources are used by Horizon Cloud to perform its operations in your Microsoft Azure subscription. See Subscription-Related Information for the Horizon Cloud Pod Deployment Wizard.

4 If you are deploying the pod with a Unified Access Gateway configuration, obtain the signed TLS/SSL server certificate that can allow your end users’ clients to trust connections to the desktops and remote applications. This certificate should match your FQDN that your end
users will use in their clients and be signed by a trusted Certificate Authority (CA). Also, all certificates in the certificate chain must have valid time frames, including any intermediate certificates. If any certificate in the chain is expired, unexpected failures can occur later in the pod onboarding process.

Unified Access Gateway presents your CA-signed certificate, so that the end users' clients can trust the connections. To support trusted access from the Internet, you deploy an external Unified Access Gateway configuration for the pod. To support trusted access within your corporate network, you use an internal Unified Access Gateway configuration. Both configuration types can be deployed during the initial pod deployment process or post-pod deployment using the Edit Pod workflow.

**Important** This FQDN cannot contain underscores. In this release, connections to the Unified Access Gateway instances will fail when the FQDN contains underscores.

5. If your signed SSL server certificate that you will use with the Unified Access Gateway configuration is not in PEM format or is not a single PEM file containing the full entire certificate chain with the private key, convert the certificate information to the required PEM format. See the steps in Convert a Certificate File to the PEM Format Required for Pod Deployment.

6. Obtain a My VMware account and register for Horizon Cloud, if you are not already registered for it.

After you have completed those preparatory tasks, log in to the console at cloud.horizon.vmware.com using your My VMware account. After logging in, you'll see the Add Cloud Capacity area on the screen and can click Add to start the pod deployment wizard. Complete the wizard by entering the required information in each screen. For detailed steps, see Deploy a Horizon Cloud Pod into Microsoft Azure.

**Note** Login authentication into the cloud-based console relies on authenticating account credentials either with the My VMware account system or VMware Cloud Services system. If those systems are experiencing a system outage and cannot take authentication requests, you will not be able to log in to the console during that time period. If you encounter issues logging in to the console's first login screen, check the Horizon Cloud System Status page at https://status.horizon.vmware.com to see the latest system status. On that page, you can also subscribe to receive updates.

**Microsoft Azure Virtual Machine Requirements for a Horizon Cloud Pod in Microsoft Azure**

Pod deployments, deployments of the pod's gateway configurations, and standard operations require specific types and sizes of virtual machines (VMs) in your Microsoft Azure cloud capacity. Your subscription needs the appropriate quotas and configuration to support these VMs. When
you are using the option to deploy the pod's external gateway in a separate subscription, that subscription needs the quota and configuration to support that external gateway configuration.

**Important** The pod deployment wizard validates that your Microsoft Azure environment has sufficient quota of cores to build the pod and the gateway configuration you specified, if any. If the wizard determines there is not sufficient quota given the subscription information you specified in the wizard, an on-screen message will display and the wizard will not proceed to its next step.

Starting with the September 2019 release's pod manifest version, both for pods newly deployed at that version and pods updated to that version, each pod has a Microsoft Azure load balancer and Microsoft Azure Database for PostgreSQL server. When a pod is updated to the September 2019 manifest or later versions, the updated pod include a Microsoft Azure load balancer and Microsoft Azure Database for PostgreSQL server. The Microsoft Azure Database for PostgreSQL server is deployed using the Single Server deployment.

**Note** GPU-enabled VMs are only available in some Microsoft Azure regions. See [Microsoft Azure Products by region](https://docs.microsoft.com/en-us/azure/virtual-machines/) for details.

In the tables below, the VM specification column provides:

- The series names that are used in the Microsoft Azure documentation
- The vCPUs family names that are used in the quotas displayed in the Microsoft Azure portal
- The specific name of the VM type from that family

To see a subscription's current quotas in the Microsoft Azure portal, navigate to **All services > Subscriptions**, click the subscription, and then click **Usage + quotas**. For more information about sizes for Microsoft Windows virtual machines in Microsoft Azure, see this topic and its subtopics in the Microsoft Azure documentation: [https://docs.microsoft.com/en-us/azure/virtual-machines/windows/sizes](https://docs.microsoft.com/en-us/azure/virtual-machines/windows/sizes).

- **Jump Box VMs**
- **Pod Manager VMs**
- **Gateway-Related VMs**
- **Golden Image VMs**
- **Farm VMs**
- **VDI Desktop VMs**

**Jump Box VMs**

Jump box VMs are temporarily created in your Microsoft Azure subscriptions for the purposes described in the tables below.
Table 2-3. Pod Jump box VM Requirements

<table>
<thead>
<tr>
<th>VM</th>
<th>Microsoft Azure VM Specification</th>
<th>Quantity</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jump box</td>
<td>Linux Standard F Family:</td>
<td>1 per pod</td>
<td>A VM created in your Microsoft Azure environment and used during the initial pod creation, and during subsequent software updates on the environment. One jump box VM for each pod you deploy. This jump box VM is deleted automatically when the pod creation or update process is finished and the VM is no longer needed.</td>
</tr>
<tr>
<td></td>
<td>Standard_F2 (2 cores, 4 GB memory)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>OS disk: Standard HDD 30 GiB</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Note** A jump box VM is newly deployed for creating a pod, for building out an update's green components when the next version of the pod software is available, for orchestrating the blue/green update process on the pod, and for the process of adding a gateway configuration to an existing pod.

Table 2-4. When Having the External Gateway in a Separate VNet: Jump box VM Requirements

<table>
<thead>
<tr>
<th>VM</th>
<th>Microsoft Azure VM Specification</th>
<th>Quantity</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jump box</td>
<td>Linux Standard F Family:</td>
<td>1 per pod</td>
<td>When optionally deploying the external gateway in its own VNet or own subscription, it needs a jump box VM separate from the jump box VM used for the pod's own core resources. This jump box VM is created in your Microsoft Azure environment in a separate resource group from the pod's jump box VM, and is used during the initial deployment of the external gateway configuration, and during subsequent software updates on that external gateway. One jump box VM for each external gateway in its own VNet or subscription that you deploy. This jump box VM is deleted automatically when the external gateway deployment or update process is finished and the VM is no longer needed.</td>
</tr>
<tr>
<td></td>
<td>Standard_F2 (2 cores, 4 GB memory)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>OS disk: Standard HDD 30 GiB</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Note** A jump box VM is newly deployed for creating one of these external gateways in its own VNet or subscription (during pod creation or when using the Edit Pod workflow to add the external gateway to an existing pod), for building out an update's green components for that external gateway when the next version of the external gateway or pod software is made available to you, for orchestrating the blue/green update process on that external gateway.

**Pod Manager VMs**

These VMs are generally considered the heart of the pod itself. The pod manager VMs are responsible for brokering the connection of the end-user clients to the Horizon agent software running in the pod-provisioned virtual desktops.
<table>
<thead>
<tr>
<th>VM</th>
<th>Microsoft Azure VM Specification</th>
<th>Quantity</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pod without high availability enabled: Pod management instances</td>
<td>Linux - Standard Dv3 Family: Standard_D4_v3 (4 cores, 16 GB memory). OS disk: Standard HDD 30 GiB. <strong>Note</strong> If the Standard_D4_v3 type is not available in your Microsoft Azure region, the deployer instead uses Standard_D3_v2 (4 cores, 14 GB memory), from the Standard Dv2 Family.</td>
<td>1 per pod during steady-state operations&lt;br&gt;2 per pod during the end-to-end time for the pod's blue/green update process.</td>
<td>For a pod without high availability enabled, during steady-state operations, one VM exists, is powered on, and runs the pod. When a new pod manifest is made available to you by VMware Operations, and the system begins building out the green components for the pod's blue/green update process, a second instance is created and powered on. As part of the end-to-end update process, you schedule the time at which the system switches to using the green components. After the switch is completed, the pod is using the newly created VM for steady-state operations and the previously used one in the blue component set is stopped and then deleted. Your environment's size must accommodate the two pod manager instances running side-by-side for the end-to-end update time period starting from the time when the system starts building out the pod's green components for the blue/green update process, to when update activities are completed and the pod is switched over to using the new green components. See the Administration Guide for a description of the pod's blue/green update process.</td>
</tr>
<tr>
<td>High-availability-enabled pod: Pod management instances</td>
<td>Linux - Standard Dv3 Family: Standard_D4_v3 (4 cores, 16 GB memory). OS disk: Standard HDD 30 GiB. <strong>Note</strong> If the Standard_D4_v3 type is not available in your Microsoft Azure region, the pod deployer instead uses Standard_D3_v2 (4 cores, 14 GB memory), from the Standard Dv2 Family.</td>
<td>2 per pod during steady-state operations&lt;br&gt;4 per pod during the end-to-end time for the pod's blue/green update process.</td>
<td>For a high-availability-enabled pod, during steady-state operations, two VMs exist, are powered on, and run the pod. When a new pod manifest is made available to you by VMware Operations, and the system begins building out the green components for the pod's blue/green update process, a second instance per pod manager VM is created and powered on. At that time, the total running pod manager VMs is four (4). As part of the end-to-end update process, you schedule the time at which the system switches to using the green components. After the switch is completed, the pod is using the two newly created VM for steady-state operations and the previously used two in the blue component set are stopped and then deleted. Your environment's size must accommodate the four pod manager instances running side-by-side for the end-to-end update time period starting from the time when the system starts building out the pod's green components for the blue/green update process, to when update activities are completed and the pod is...</td>
</tr>
</tbody>
</table>
Table 2-5. Pod Management VM Requirements - For the Pod’s Core VMs, Not Including Gateway Configurations (continued)

<table>
<thead>
<tr>
<th>VM</th>
<th>Microsoft Azure VM Specification</th>
<th>Quantity</th>
<th>Description</th>
</tr>
</thead>
</table>

switched over to using the new green components. See the Administration Guide for a description of the pod’s blue/green update process.

Gateway-Related VMs

Gateway-related VMs are:

- The Unified Access Gateway instances configured to function as secure gateways for the end-user clients accessing the pod-provisioned resources.
- When you deploy the external gateway into a separate VNet, the gateway connector VM that handles the cloud management operations on that external gateway configuration.

**Note** Starting with the July 2020 quarterly release, you can choose from a list of supported VM models for the Unified Access Gateway instances when you are deploying a brand new gateway — either at the time of deploying the whole pod or when adding a new gateway. Prior to the July 2020 release, the gateway instances were required to use the Standard_A4_v2 VM model. The list of supported VM models that you can choose from in the on-screen wizard will depend on which VM models are available in the Microsoft Azure region into which you are deploying the gateway instances. The displayed choices will also depend on your VM quota in the Microsoft Azure subscription that you are using for the gateway deployment. The pod deployment wizard’s **VM Model** menu will dynamically reflect the VM models that meet those requirements.

Software updates will maintain the gateway instances’ VM models. Whatever the gateway instances’ VM model is prior to a pod update will be their VM model after the update.
Unified Access Gateway is an optional feature that is deployed for your pod when you configure the gateway settings in the deployment wizard. If you decide to have Unified Access Gateway instances for the pod, your environment must accommodate these instances running during the pod’s end-to-end blue/green update process. The number of steady-state instances depends on whether you choose to have both external and internal Unified Access Gateway configurations.

When you have only an external or only an internal Unified Access Gateway configuration, during steady-state operations, two instances exist, are powered on, and provide the Unified Access Gateway capabilities. During an update process, two additional instances are created and powered on to run the software updates on Unified Access Gateway. After the update is completed, the pod migrates to using the newly created VMs and the previously used ones in the blue component set are stopped and then deleted.

When you have both internal and external Unified Access Gateway configurations, during steady-state operations, four instances exist, are powered on, and provide the Unified Access Gateway capabilities. Two instances provide the capabilities for the external configuration and two instances provide the capabilities for the internal configuration. During an update process, two additional instances per configuration are created and powered on to run the software updates on Unified Access Gateway. After the update is completed, the pod migrates to using the newly created VMs and the previously used ones in the blue component set are stopped and then deleted.

Your environment’s size must accommodate the indicated Unified Access Gateway instances running side-by-side for the end-to-end update time period starting from the time when the system starts building out the pod’s green components for the blue/green update process, to when update activities are

### Table 2-6. Unified Access Gateway VM Requirements

<table>
<thead>
<tr>
<th>VM</th>
<th>Microsoft Azure VM Specification</th>
<th>Quantity</th>
<th>Description</th>
</tr>
</thead>
</table>
| Unified Access Gateway instances | Starting in this release, you can choose from the following VM models for new gateway deployments.  
- Linux Standard Av2 Family  
  - Standard_A4_v2 (4 cores, 8 GB memory), OS disk: Standard HDD 20 GiB  
- Linux Standard FSv2 Family:  
  - Standard_F8s_v2 (8 cores, 16 GB memory), OS disk: SSD 32 GiB | Varied based on whether you choose to have an external or internal Unified Access Gateway configuration, or both types for the same pod.  
For an external-only or an internal-only configuration:  
- 2 per pod during steady-state operations  
- 4 per pod during the end-to-end time for pod-related blue/green update activities. For a pod with both an external and internal Unified Access Gateway configuration,  
- 4 per pod during steady-state operations  
- 8 per pod during the end-to-end time for pod-related blue/green update activities. | Unified Access Gateway is an optional feature that is deployed for your pod when you configure the gateway settings in the deployment wizard. If you decide to have Unified Access Gateway instances for the pod, your environment must accommodate these instances running during the pod’s end-to-end blue/green update process. The number of steady-state instances depends on whether you choose to have both external and internal Unified Access Gateway configurations. When you have only an external or only an internal Unified Access Gateway configuration, during steady-state operations, two instances exist, are powered on, and provide the Unified Access Gateway capabilities. During an update process, two additional instances are created and powered on to run the software updates on Unified Access Gateway. After the update is completed, the pod migrates to using the newly created VMs and the previously used ones in the blue component set are stopped and then deleted. When you have both internal and external Unified Access Gateway configurations, during steady-state operations, four instances exist, are powered on, and provide the Unified Access Gateway capabilities. Two instances provide the capabilities for the external configuration and two instances provide the capabilities for the internal configuration. During an update process, two additional instances per configuration are created and powered on to run the software updates on Unified Access Gateway. After the update is completed, the pod migrates to using the newly created VMs and the previously used ones in the blue component set are stopped and then deleted. Your environment’s size must accommodate the indicated Unified Access Gateway instances running side-by-side for the end-to-end update time period starting from the time when the system starts building out the pod’s green components for the blue/green update process, to when update activities are |
Table 2-6. Unified Access Gateway VM Requirements

<table>
<thead>
<tr>
<th>VM</th>
<th>Microsoft Azure VM Specification</th>
<th>Quantity</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>completed and the pod is switched over to using the new green components. See the Administration Guide for a description of the pod’s blue/green update process.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 2-7. When Having the External Gateway in a Separate VNet: Gateway Connector VM Requirements

<table>
<thead>
<tr>
<th>VM</th>
<th>Microsoft Azure VM Specification</th>
<th>Quantity</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gateway Connector instance</td>
<td>Linux Standard Av2 Family: Standard_A1_v2 (1 cores, 2 GB memory) OS disk: Standard HDD 10 GiB</td>
<td>1 per this external-gateway type during steady-state operations 2 per this external-gateway type during the end-to-end time for pod-related blue/green update activities.</td>
<td>When the external gateway is deployed into a separate VNet, this VM is created and used for the cloud management operations on that external gateway configuration. During an update process, an additional instance is created and powered on to run the software updates on the Unified Access Gateway in the external gateway configuration. After the update is completed, migration to the newly created VM occurs and the previously used one is stopped and then deleted. If you decide to use this optional configuration, your environment must accommodate these instances running end-to-end during the pod-related blue/green update activities.</td>
</tr>
</tbody>
</table>

Golden Image VMs

A golden image is a Microsoft Windows operating system VM that is configured so that Horizon Cloud can convert it into a published image. Sometimes you’ll see these VMs referred to as gold patterns.
<table>
<thead>
<tr>
<th>VM</th>
<th>Microsoft Azure VM Specification</th>
<th>Quantity</th>
<th>Description</th>
</tr>
</thead>
</table>
| Golden images       | For GPU-enabled golden images, the system uses:                                         | Varied   | A golden image is a Microsoft Windows operating system VM that is configured so that Horizon Cloud can convert it into a published image. An RDSH-capable Windows operating system VM provides the base used to create the RDSHs in farms that provide session-based desktops and remote applications to your end users. A Windows client operating system VM provides the base used to create the VDI desktops. Each golden image is a combination of Microsoft Windows operating system and whether it is GPU-enabled or not. So if you want your pod to provide:   
|                     |   - The NV-series NV6 Standard (from the Standard NV Family vCPUs)                      | based on | RDSH desktops using Microsoft Windows 2016 Datacenter, no GPU   
|                     |   - OS disk: Standard HDD 127 GiB                                                     | your needs | RDSH desktops using Microsoft Windows 2016 Datacenter, with GPU   
|                     | For non-GPU-enabled golden images and Microsoft Windows client operating systems, the system uses: |          | Then you need at least 2 golden image VMs.   
|                     |   - The Standard_D4_v3 (from the Standard Dv3 Family vCPUs)                           |          | The process of converting a golden image into a published image is sometimes called publishing the image, or also called sealing the image. The resulting published image is sometimes called a sealed image or an assignable image, because it is in a finalized state for use in assignments.   
|                     |   - OS disk: Standard HDD 127 GiB                                                     |          | The system automatically powers off the golden image when it is published (when you perform the **Convert to Image** action on the golden image in the administrative console). When you update a published image, the system powers the VM on again.   
|                     | If Microsoft does not provide the Standard Dv3 Family in the Microsoft Azure region where you deployed the pod, the system uses the Standard_D3_v2 instead from the Standard Dv2 Family. |          | **Note** When you duplicate an image using the console, the system temporarily powers on the golden image’s VM to obtain its configuration for the duplicate, and then powers it off again.   
|                     | For non-GPU-enabled golden images and Microsoft Windows RDSH-capable operating systems, the system uses: |          | For information about how to create a golden image, see the topic Creating Desktop Images for a Horizon Cloud Pod in Microsoft Azure in the Administration Guide.   
|                     |   - The Standard_D2_v3 (from the Standard Dv3 Family vCPUs)                           |          |   
|                     |   - OS disk: Standard HDD 127 GiB                                                     |          |   
|                     | If Microsoft does not provide the Dv3-series in the Microsoft Azure region where you deployed the pod, the system uses the Standard_D2_v2 instead from the Standard Dv2 Family. |          |   
| Golden images       | For non-GPU-enabled golden images and Microsoft Windows RDSH-capable operating systems, the system uses: |          |   
| are either GPU-      |   - The Standard_D4_v3 (from the Standard Dv3 Family vCPUs)                           |          |   
| enabled or not,     |   - OS disk: Standard HDD 127 GiB                                                     |          |   
| dependin            | For non-GPU-enabled golden images and Microsoft Windows RDSH-capable operating systems, the system uses: |          |   
| g on your           |   - The Standard_D2_v3 (from the Standard Dv3 Family vCPUs)                           |          |   
| selection when you  |   - OS disk: Standard HDD 127 GiB                                                     |          |   
| create them.        | For non-GPU-enabled golden images and Microsoft Windows RDSH-capable operating systems, the system uses: |          |   
|                     |   - The Standard_D4_v3 (from the Standard Dv3 Family vCPUs)                           |          |   
|                     |   - OS disk: Standard HDD 127 GiB                                                     |          |   
|                     | If Microsoft does not provide the Dv3-series in the Microsoft Azure region where you deployed the pod, the system uses the Standard_D2_v2 instead from the Standard Dv2 Family. |          |   

Horizon Cloud Deployment Guide
Farm VMs

RDSH farm VMs are the RDS-capable instances that provide session-based desktops and remote applications to your end users. You need at least one RDSH farm to deliver session desktops and one RDSH farm to deliver remote applications. To meet administrator or end-user needs, you can decide to deploy additional farms.

**Note** In the current service release, you cannot deliver both session-based desktops and remote applications from the same farm.
Table 2-9. Farm VM Requirements

<table>
<thead>
<tr>
<th>VM</th>
<th>Microsoft Azure VM Specification</th>
<th>Quantity</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>RDSH farm</td>
<td>You can customize the set of Microsoft Azure VM types that you want available for selection when creating farms in your pod. You can customize your own list from the set of Microsoft Azure VM sizes that are generally available in the standard Microsoft Azure regions. For more information about customizing the set of VM types available to use in your farms, see the Horizon Cloud Administration Guide. When creating or editing a farm, you can customize the OS disk size of the farm's RDSH instances to change it from the system default value. For specific details about the Windows VM sizes that are generally available in the standard Microsoft Azure regions, see the Microsoft documentation at <a href="https://docs.microsoft.com/en-us/azure/virtual-machines/windows/sizes">https://docs.microsoft.com/en-us/azure/virtual-machines/windows/sizes</a>. Note For production environments, ensure the VM types you use for your farms have a minimum of two (2) CPUs. Meeting this criteria avoids unexpected end-user connection issues. This criteria is a result of the Horizon Agent recommendations to have a minimum of 2 CPUs to install or update Horizon Agent from version 7.x or later. This Horizon Agent criteria is stated in the Horizon product documentation starting at version 7.8 (the references to a 2 CPU minimum starts with this version 7.8 of Install Horizon Agent on a Virtual Machine).</td>
<td>Varied, based on your needs and how you have customized the VM sizes in your Horizon Cloud environment.</td>
<td>The power state of these VMs varies, depending on the farm configuration settings and the end-user demand.</td>
</tr>
</tbody>
</table>
VDI Desktop VMs
VDI desktop VMs are the instances that provide VDI desktops to your end users.

**Note**  A new feature in the July 2020 quarterly service release is the use of App Volumes features with pods in Microsoft Azure. When you use the console's App Volumes capture process to add native applications to your Horizon Cloud inventory, the system creates a VDI desktop assignment of two VMs to support the capture process. The VM type used for this system-generated assignment is the same model as is used for the published image you selected in the console for the application capture process.
<table>
<thead>
<tr>
<th>VM</th>
<th>Microsoft Azure VM Specification</th>
<th>Quantity</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>VDI desktops</td>
<td>You can customize the set of Microsoft Azure VM types that you want available for selection when creating VDI desktop assignments in your pod. You can customize your own list from the set of Microsoft Azure VM sizes that are generally available in the standard Microsoft Azure regions. For more information about customizing the set of VM types available to use in your VDI desktop assignments, see the Horizon Cloud Administration Guide.</td>
<td>Varied, based on your needs and how you have customized the VM sizes in your Horizon Cloud environment.</td>
<td>The power state of these VMs varies, depending on the VDI desktop assignment settings and the end-user demand.</td>
</tr>
</tbody>
</table>

**Note** A small set of Microsoft Azure VM sizes that Microsoft has determined are not appropriate for VDI use cases are automatically omitted from use, such as Standard_B2ls and Standard_B1s.

When creating or editing a VDI desktop assignment, you can customize the OS disk size of the VDI desktop instances to change it from the system default.
## Table 2-10. VDI Desktop VM Requirements

<table>
<thead>
<tr>
<th>VM</th>
<th>Microsoft Azure VM Specification</th>
<th>Quantity</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>For specific details about those Windows VM sizes, see the Microsoft documentation at <a href="https://docs.microsoft.com/en-us/azure/virtual-machines/windows/sizes">https://docs.microsoft.com/en-us/azure/virtual-machines/windows/sizes</a>.</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Note** For production environments, ensure the VM types you use for your VDI desktop assignments have a minimum of two (2) CPUs. Meeting this criteria avoids unexpected end-user connection issues. This criteria is a result of the Horizon Agent recommendations to have a minimum of 2 CPUs to install or update Horizon Agent from version 7.x or later. This Horizon Agent criteria is stated in the Horizon product documentation starting at version 7.8 (the references to a 2 CPU minimum starts with this [version 7.8 of Install Horizon Agent on a Virtual Machine](https://docs.vmware.com/en/horizon/7.8/topic/com.vmware.horizon.doc/doc/index.html#Install_Horizon_Agent_on_a_Virtual_Machine)).

## VMware Horizon Cloud Service on Microsoft Azure Service Limits

This topic describes some of the common VMware Horizon Cloud Service on Microsoft Azure limits, which are also called supported maximums. This topic currently describes the supported maximums on both the number of desktop and farm RDSH VMs you can deploy in a single subscription and on the total number of concurrent connected sessions you can have per pod. Over time, this topic will be updated to list more of the known limits.

The service is tested up to a certain number of deployed VMs in a single subscription and the number of simultaneous connections that a pod can accommodate.

### Maximum of 2,000 desktop VMs and farm RDSH VMs per subscription

This limit is based on Microsoft Azure API limits that are given on a single subscription. To work well within these API limits during standard operations, Horizon Cloud supports up to a maximum of 2,000 desktop VMs and farm RDSH VMs per subscription.
The 2,000 number per subscription includes VDI desktop VMs and farm RDSH VMs, and applies across all pods in the single subscription. For example, if you have one pod in your subscription, you might have up to 2,000 VDI desktops on that pod, or 1,950 VDI desktops plus 50 farm RDSH VMs. If you have more than one pod in your subscription, the number of VDI desktops and farms RDSH VMs across all of the pods must not total more than 2,000.

Maximum of 2,000 sessions per pod

Horizon Cloud supports running up to 2,000 concurrent connected sessions per pod. That 2,000 number includes connections to VDI desktops, RDS desktops, and RDS applications served by the pod. The pod's session-handling capabilities determine this maximum.

Configure the Required Virtual Network in Microsoft Azure

Your Microsoft Azure environment must have an existing virtual network before you can deploy the Horizon Cloud pod into the environment. If you do not already have a virtual network (VNet) in the region into which you are deploying, you must create the virtual network. If you want to have the pod's external gateway deployed into its own VNet — separate from the pod's VNet, you must create that VNet also and then peer the two VNets. If you want to have the pod's external gateway using its own subscription, separate from the pod's, then you must create a separate VNet to use for that external gateway in that subscription and peer the two VNets. Because a single VNet does not span subscriptions, choosing to deploy an external gateway into its own subscription also will require the external gateway to use a VNet that is separate from and peered with the pod's VNet.

Caution The subnets you manually create on your VNet in advance for the pod deployment must remain empty. Do not put any resources on these subnets or otherwise use any of the IP addresses. If an IP address is already in use on the subnets, the pod might fail to deploy.
### Into which VNet are you deploying the external gateway?

<table>
<thead>
<tr>
<th>Subnet creation</th>
<th>Subnets needed</th>
</tr>
</thead>
<tbody>
<tr>
<td>When deploying a pod with the external gateway using the pod’s VNet</td>
<td>For this configuration, you can either create subnets in advance on the VNet and specify those in the pod deployment wizard, or directly type into the wizard the address spaces for the needed subnets and the pod deployer will create the subnets in the VNet.</td>
</tr>
<tr>
<td></td>
<td><strong>Important</strong> If your existing VNet is peered, the deployer cannot automatically update the VNet’s address space. If the VNet is peered, the best practice is to create the subnets in advance as described in <em>In Advance of Pod Deployment, Create the Horizon Cloud Pod’s Required Subnets on your VNet in Microsoft Azure</em>. If you do not want to create the subnets in advance and you enter subnet CIDRs in the deployment wizard that are not contained within the VNet’s existing address space, the wizard will display an error message and you will need to specify valid subnet address spaces to proceed, or use an unpeered virtual network.</td>
</tr>
<tr>
<td></td>
<td>Pod deployment using this configuration requires following subnets:</td>
</tr>
<tr>
<td></td>
<td>- <strong>Management subnet</strong>, for IP addresses used by the VMs involved in management activities of the pod itself</td>
</tr>
<tr>
<td></td>
<td>- <strong>Primary VM subnet</strong> — also known as the tenant subnet or the desktop subnet. This subnet provides IP addresses used for the RDSH server VMs and VDI desktop VMs on that subnet. When the internal Unified Access Gateway configuration is specified in the...</td>
</tr>
<tr>
<td>Into which VNet are you deploying the external gateway?</td>
<td>Subnet creation</td>
</tr>
<tr>
<td>-------------------------------------------------------</td>
<td>-----------------</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td>Into which VNet are you deploying the external gateway?</td>
<td>Subnet creation</td>
</tr>
<tr>
<td>--------------------------------------------------------</td>
<td>----------------</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

When deploying a pod with the choice to have the external gateway using its own VNet or subscription, separate from the pod's VNet or subscription

For this configuration, because there are two VNets involved and these VNets must be peered, the best practice is to create the subnets in advance on the VNet and specify those in the pod deployment wizard. Create the subnets in advance as described in In Advance of Pod Deployment, Create the Horizon Cloud Pod's Required Subnets on your VNet in Microsoft Azure. Even though the deployment wizard gives you the option of directly typing into the wizard the address spaces for the needed subnets to have the deployer create the subnets for you, if you specify address spaces that are not already in the VNet's address space, the deployer will not be able to add them to the VNet because it is a peered VNet.

In this case, one VNet will have the subnets for the pod and one VNet will have the subnets for the external gateway. Those two VNets must be peered. Let's refer to the pod's VNet as VNet-1 and the external gateway's VNet as VNet-2. For each of these VNets, you can either specify the

In this type of deployment, the pod's VNet (VNet-1) gets a management subnet and a desktop subnet, used for the same purposes as described for when the external gateway is in the pod's own VNet. However, the pod's VNet does not get a DMZ subnet in this configuration because the DMZ subnet would be used by the external Unified Access Gateway configuration, which is in the other VNet (VNet-2) in this configuration. In this deployment configuration, the external gateway’s VNet gets the following subnets:

■ Management subnet, for IP addresses used by the VMs involved in management activities of the external gateway itself (the temporary jump box, the gateway's connector VM, and the external gateway’s Unified Access Gateway instances)

■ Back-end subnet used by the external gateway’s Unified Access Gateway instances

■ DMZ subnet used by the external gateway’s Unified Access Gateway instances
Into which VNet are you deploying the external gateway?

<table>
<thead>
<tr>
<th>Subnet creation</th>
<th>Subnets needed</th>
</tr>
</thead>
<tbody>
<tr>
<td>address spaces for the subnets that the pod deployer will automatically create or specify subnets that you have created in advance.</td>
<td></td>
</tr>
</tbody>
</table>

You perform these steps using the Microsoft Azure portal appropriate for your registered account. For example, there are specific portal endpoints for these Microsoft Azure clouds:

- Microsoft Azure (standard global)
- Microsoft Azure China
- Microsoft Azure US Government

Log in to the portal using the URL appropriate for your account.

Procedure

1. From the Microsoft Azure portal's left navigation bar, click **Virtual networks** and then click **Add**.

   The **Create virtual network** wizard appears, displaying its Basics step.

2. Follow the wizard's on-screen steps to specify the following information.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Subscription</td>
<td>Select the same subscription that you are planning to use when you deploy the pod.</td>
</tr>
<tr>
<td>Resource Group</td>
<td>You can either choose an existing resource group or have a new one created when the virtual network is created.</td>
</tr>
<tr>
<td>Name</td>
<td>Specify a name for the VNet.</td>
</tr>
<tr>
<td>Region</td>
<td>Select the same Microsoft Azure region into which you are planning to deploy the pod.</td>
</tr>
<tr>
<td>Address space</td>
<td>Specify the VNet's address space.</td>
</tr>
<tr>
<td>Subnet and Address range</td>
<td>Microsoft Azure requires creating one subnet when creating a VNet. You can either retain the default values or customize the name and range. If you want to use this subnet for one of the pod's required subnets, specify the appropriate address range according to the pod deployer requirements. As an example, if you want to use this subnet for the pod's tenant subnet, ensure it has an IP address range to match the /27 minimum that the deployment wizard requires. See In Advance of Pod Deployment, Create the Horizon Cloud Pod's Required Subnets on your VNet in Microsoft Azure.</td>
</tr>
</tbody>
</table>

   **Important** If you use this subnet for one of the pod's required subnets, you cannot also use it for other resources.

3. Retain the default values for the optional settings.

4. Proceed to the review step and then click **Create**.
Results

The virtual network (VNet) is created in your Microsoft Azure account.

What to do next

If you are manually creating the required subnets instead of having the pod deployment process create them, configure the newly created VNet with the subnets you will use for the pod. See the steps in In Advance of Pod Deployment, Create the Horizon Cloud Pod's Required Subnets on your VNet in Microsoft Azure and When Using Existing Subnets for a Horizon Cloud Pod in Microsoft Azure.

Configure the newly created VNet with a working DNS service and connectivity to the Active Directory service you will use with your pod. See the steps in Configure the DNS Server Settings Needed by the VNet Topology You Will Use for Your Horizon Cloud Pods in Microsoft Azure.

Ensure your VNet configuration, in terms of your firewalls and other network behavior, adheres to the pod deployment DNS, ports, and protocols requirements described in DNS Requirements for a Horizon Cloud Pod in Microsoft Azure and Ports and Protocols Requirements for a Horizon Cloud Pod at the September 2019 Release's Manifest or Later.

**Important** The pod's temporary jump box VM and pod's manager VM require outbound Internet access on your Microsoft Azure VNet. If you are deploying an external gateway in its own VNet, that VNet must support the external gateway's temporary jump box and gateway connector VM having outbound Internet access. If you require proxy-based outbound Internet access, you will need to specify the proxy server information as you complete the fields in the pod deployment wizard.

**In Advance of Pod Deployment, Create the Horizon Cloud Pod's Required Subnets on your VNet in Microsoft Azure**

If you are using a peered VNet, a best practice is to create the required subnets in advance of deploying the pod, to ensure that you have accounted for the address spaces your subnets need in the VNet prior to running the deployment wizard. Even when your VNet is not peered, instead of having the pod deployment process create the required subnets, you can create them in advance on your VNet.

**Important** Starting with the September 2019 release's pod manifest version, both for pods newly deployed at that manifest version or later and for pods updated to that version or later versions, the pod's management subnet must also support network communication with the pod's Microsoft Azure Database for PostgreSQL service resource. Before deploying a new pod or upgrading an existing pod, the pod management subnet that you create must have the Microsoft.Sql service listed as a service endpoint. The deployment or update process will check if the subnet has the endpoint and not proceed if the endpoint is not enabled on the subnet. For details, see When Using Existing Subnets for a Horizon Cloud Pod in Microsoft Azure.
When you create the subnets in advance, you must ensure their address ranges, in classless interdomain routing (CIDR) notation, adhere to the pod deployment wizard's minimum requirements:

- For the management subnet, a CIDR of /27 or more is required. This subnet is for IP addresses used by the VMs involved in management activities of the pod itself.

- For the primary VM subnet — also known as the desktop or tenant subnet — a CIDR of /27 or more is required. For production environments, a CIDR of /24 to /21 is recommended (256 addresses to 2048 addresses). This subnet is for IP addresses used for the RDSH server VMs and VDI desktop VMs on that subnet. The pod's manager VM uses an IP address from this subnet. If the pod will have an internal Unified Access Gateway configuration, those Unified Access Gateway VMs also use IP addresses from this subnet. If the pod will have an external gateway configuration that is deployed using the pod’s VNet, that external gateway’s Unified Access Gateway VMs also use IP addresses from this subnet.

**Important** The VMs for your VDI desktops, the RDS-capable images, and every RDSH VM in the pod's farms consume these IP addresses. Because this primary VM subnet cannot be extended after the pod is deployed, ensure you set this range large enough to accommodate the number of desktops you anticipate you might want this pod to provide. For example, if you anticipate this pod should provide over 1000 desktops in the future, ensure this range provides for more than that number of IP addresses. Starting in the July 2020 release, a new feature allows you to later edit the pod and add additional VM subnets for use by your farm VMs and VDI desktop VMs. That new feature gives you the flexibility to add VM subnets over time to accommodate growth in your farms and VDI desktop assignments. Because the system will default to using this primary VM subnet unless you expressly specify those additional subnets in the definitions of your farms and VDI desktop assignments, it is a best practice to ensure the range for this primary VM subnet to be large enough to accommodate your anticipated number of farm VMs and desktops.

- If you are going to have an external Unified Access Gateway configuration deployed into the pod’s VNet, you need a DMZ subnet, with a CIDR of /28 or more. This subnet is for IP addresses used by the Unified Access Gateway VMs' NICs to communicate with this external gateway configuration's load balancer. If you want to keep the management and DMZ subnet ranges co-located, you could specify the DMZ subnet range similar to the management subnet with an IP specified. For example, if the management subnet is 192.168.8.0/27, a matching DMZ subnet would be 192.168.8.32/27.

- If you are going to have the external Unified Access Gateway configuration deployed into its own VNet, separate from the pod’s, that VNet needs three subnets:
  - A management subnet, of a CIDR of /27 more is required. This subnet is for IP addresses used by the VMs involved in management activities of the external gateway overall, such as the gateway connector VM.
  - A back-end subnet, of a CIDR of /27 more is required. This subnet is for IP addresses used by the Unified Access Gateway VMs' NICs to communicate with the pod-provisioned farm and desktop VMs over the peered VNet with the pod's VNet.
A front-end (DMZ) subnet, of a CIDR of /28 or more. This subnet is for IP addresses used by the Unified Access Gateway VMs' NICs to communicate with the external gateway's load balancer. If you want to keep the management and front-end subnet ranges co-located in this VNet, you could specify the DMZ subnet range similar to the management subnet with an IP specified. For example, if the management subnet is 192.168.8.0/27, a matching front-end subnet would be 192.168.8.32/27.

**Important** For each CIDR, ensure that each combination of prefix and bit mask results in an IP address range having the prefix as the starting IP address. Microsoft Azure requires that the CIDR prefix be the start of the range. For example, a correct CIDR of 192.168.182.48/28 would result in an IP range of 192.168.182.48 to 192.168.182.63, and the prefix is the same as the starting IP address (192.168.182.48). However, an incorrect CIDR of 192.168.182.60/28 would result in an IP range of 192.168.182.48 to 192.168.182.63, where the starting IP address is not the same as the prefix of 192.168.182.60. Ensure that your CIDRs result in IP address ranges where the starting IP address matches the CIDR prefix.

**Prerequisites**

Ensure your Microsoft region has the VNet that you plan to use for your pod. See Configure the Required Virtual Network in Microsoft Azure.

Ensure the address ranges you plan to use for the subnets do not overlap. The pod deployment wizard will display an error if the subnet ranges overlap.

**Procedure**

1. In the Microsoft Azure portal, navigate to the VNet for which you need to create the described subnets.
2. Click **Subnets**.
3. Click **+ Subnet**.
   - The **Add subnet** screen appears.
4. Provide the information for the required fields.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Specify a name for the subnet.</td>
</tr>
<tr>
<td>Address range (CIDR block)</td>
<td>Type a CIDR for the subnet.</td>
</tr>
</tbody>
</table>

5. If this subnet is going to be the management subnet, in the **Service endpoints** section, select the Microsoft.Sql service.
6. Click **OK**.
   - The subnet is added to the VNet.
7. Repeat steps 3 through 5 to add the remaining required subnets.
8 If you are going to deploy the external gateway in its own VNet, repeat the steps for that VNet's subnets.

Results

**Caution** The subnets you manually create on your VNet in advance for the pod deployment must remain empty. Do not put any resources on these subnets or otherwise use any of the IP addresses. If an IP address is already in use on the subnets, the pod might fail to deploy.

What to do next

For any management subnets you created, ensure the Microsoft.Sql service is enabled as a service endpoint. See When Using Existing Subnets for a Horizon Cloud Pod in Microsoft Azure. This service must be enabled on the pod's management subnet, and if you are deploying the external gateway in its own VNet, the service must be enabled on that gateway's management subnet also.

**When Using Existing Subnets for a Horizon Cloud Pod in Microsoft Azure**

Starting with the September 2019 release, both for pods newly deployed using that release's manifest version or later versions, and for pods updated to that release's manifest version or later versions, a pod's management subnet must also support network communication with the Microsoft Azure Database for PostgreSQL service endpoint. Before deploying a new pod or upgrading an existing pod, the pod management subnet that you create must have the Microsoft.Sql service enabled as a service endpoint. The deployment or update process will check if the subnet has the endpoint and not proceed if the endpoint is not enabled on the management subnet. In addition to enabling that service endpoint, if you have firewall or network security group (NSG) rules on your management subnet, you must configure it to allow traffic for the Microsoft Azure Database for PostgreSQL service before deploying a new pod or upgrading an existing pod.

**Important** The December 2019 release debuted the feature to deploy the pod's external Unified Access Gateway configuration into its own VNet, separate from the pod's VNet. When using that feature, the management subnet in the external gateway's VNet must also adhere to this requirement to have the Microsoft.Sql service enabled as a service endpoint on that subnet.

The September 2019 release introduced use of the Microsoft Azure Database for PostgreSQL service as a required element of a Horizon Cloud pod in Microsoft Azure. As described in the Microsoft documentation, Microsoft Azure Database for PostgreSQL is a fully managed database-as-a-service offering. In a pod deployment or update, a Microsoft Azure Database for PostgreSQL server resource is deployed in the pod's resource group, using the Single Server type of deployment. The deployment and update processes also automatically add a VNet rule to the pod's VNet. This VNet rule restricts the Microsoft Azure Database for PostgreSQL server's traffic to the pod's management subnet. Communication between the pod and that Microsoft Azure Database for PostgreSQL server use the management subnet, which places some requirements on the pod's management subnet.
On the Management Subnet, Enable the Microsoft.Sql Service as a Service Endpoint

The VNet rule to restrict traffic for the deployed Microsoft Azure Database for PostgreSQL server to the management subnet requires the subnet to have the Microsoft.Sql service endpoint enabled. In the scenario where you have the pod deployer create the subnets, the deployer ensures the pod’s management subnet has the Microsoft.Sql service endpoint enabled on the management subnet that it creates. However, when you create the management subnet yourself, you must ensure that management subnet meet these requirements before you deploy a new pod or update an existing pod. The following screenshot is an example to illustrate where you enable the Microsoft.Sql service as a service endpoint on a subnet using the Microsoft Azure portal. After clicking on the subnet in the portal, in the **Service endpoints** section, use the **Services** drop-down list to select Microsoft.Sql, and then save.

You can use the Microsoft Azure portal to navigate to the management subnet and select Microsoft.Sql in the **Services** drop-down.

<table>
<thead>
<tr>
<th>Address range (CIDR block)</th>
<th>192.168.24.0/27</th>
</tr>
</thead>
<tbody>
<tr>
<td>Available addresses</td>
<td>17</td>
</tr>
<tr>
<td>Network security group</td>
<td>None</td>
</tr>
<tr>
<td>Route table</td>
<td>None</td>
</tr>
<tr>
<td>Service endpoints</td>
<td>Microsoft.Sql</td>
</tr>
<tr>
<td>Service endpoint status</td>
<td>Succeeded</td>
</tr>
<tr>
<td>Subnet delegation</td>
<td>None</td>
</tr>
</tbody>
</table>
Ensure Your Firewalls or NSGs Allow for Pod Communication to the Microsoft Azure Database for PostgreSQL Service

As listed in DNS Requirements for a Horizon Cloud Pod in Microsoft Azure, on the management subnet, you must configure your network rules for the management subnet to allow communication from the pod to the Microsoft Azure Database for PostgreSQL service. You must ensure your management subnets meets this requirement before you deploy a new pod or update an existing pod.

If your firewalls or NSGs support using service tags to specify access, allow pod communication with one of the following:

- Global Azure SQL service tag: Sql
- Region-specific SQL service tag for the Azure region where the pod is deployed: Sql.region, such as Sql.WestUS.

If your firewalls or NSGs do not support using service tags to specify access, you can use the host name of the database server resource that is created in the pod's resource group. The server resource's name follows the pattern *.postgres.database.azure.com.

For information about service tags in security groups, see the Microsoft Azure documentation topic at Service tags.

Configure the DNS Server Settings Needed by the VNet Topology You Will Use for Your Horizon Cloud Pods in Microsoft Azure

The VNets that your Horizon Cloud pods are deployed into must have the ability to resolve both internal machine names and external names. During the pod deployment process, the deployer securely downloads the pod software into your Microsoft Azure environment from external addresses in the Horizon Cloud control plane. The ability to resolve internal virtual machine (VM) names is needed for the pod's Horizon Cloud Active Directory domain-join operations with the VMs that get deployed in your Microsoft Azure environment.

**Important** Ultimately, the key requirement is that the pod-related VMs that need to reach specific DNS names are able to do that. That you have your VNet topology configured so that both internal machine names and external names can be resolved by the pod-related VMs that need to do that. You need to ensure that whatever VNet topology you are using in Microsoft Azure that you want to deploy the pod into will allow the pod VMs deployed onto the relevant required subnets can get that DNS name resolution. For specifics on the DNS resolution requirements, see DNS Requirements for a Horizon Cloud Pod in Microsoft Azure.

If you are going to use the feature to have the external gateway deployed into its own VNet separate from the pod's VNet, those VNets must be peered and you and your networking team must ensure that your peered VNets topology provides for that topology's DNS settings to meet the pod's key DNS requirements, as described in the previous paragraph. The Horizon Cloud documentation set is not covering details of such advanced VNet topologies that your networking team might have customized for your use.
In a Microsoft Azure subscription, internal network connectivity is not set up by default. For production environments, you or your networking team would typically configure the virtual network's DNS settings to point at a valid DNS server that can resolve external names as well as work in Microsoft Azure for your corporate machines. For example, you might want to deploy a Microsoft Windows Server 2016 virtual machine in that virtual network to act as the DNS server, and configure the virtual network's DNS setting to point to the IP address of that deployed DNS server.

For proof-of-concept environments, if your organization's privacy and security policies allow, you can configure the internal DNS to delegate to an external public DNS for external name resolution. Some organizations and ISPs provide public recursive name servers to use for such purposes, such as OpenDNS at 208.67.222.222 or Google Public DNS at 8.8.8.8. For a sample list of public recursive name servers, see the Wikipedia article Public recursive name server.

Prerequisites

Ensure your Microsoft Azure region has the VNet topology that you plan to specify in the pod deployer wizard. See Configure the Required Virtual Network in Microsoft Azure.

Ensure that the DNS server settings that you or your networking team are going to configure for that VNet topology can reach and resolve the specific external names required for a successful pod deployment. For details, see DNS Requirements for a Horizon Cloud Pod in Microsoft Azure.

Procedure

1. From the Microsoft Azure portal's left navigation bar, click Virtual networks and then click the virtual network that you are going to use for your pod.

2. Display the virtual network's DNS server settings by clicking DNS servers.
3. Using the Custom option, add the address of the DNS server you want to use for name resolution and click Save.

What to do next

Ensure the pod deployer's access requirements for DNS, ports, and protocols are met from your VNet topology. See DNS Requirements for a Horizon Cloud Pod in Microsoft Azure and Ports and Protocols Requirements for a Horizon Cloud Pod at the September 2019 Release's Manifest or Later.

Active Directory Domain Configurations

A Horizon Cloud environment requires registering at least one Active Directory (AD) domain with the Horizon Cloud pod. This topic describes the configurations that are supported for use with your Horizon Cloud pods in Microsoft Azure.

The supported configurations are:

- On-premises AD server and connecting that on-premises AD with your Microsoft Azure environment using VPN/MPLS or Microsoft Azure Express Route.
- AD server running in your Microsoft Azure environment.
- Using Microsoft Azure Active Directory Domain Services. For an overview of these services that Microsoft Azure provides, see this Azure AD Domain Services article in the Microsoft documentation.

For an in-depth technical description of each supported configuration, some options for each, and the advantages and disadvantages of each, see the VMware technical paper Networking and Active Directory Considerations on Microsoft Azure with VMware Horizon Cloud.

Important Your Horizon Cloud environment can consist of pods in Microsoft Azure and Horizon pods on-premises and in VMware Cloud on AWS. As a result, all of those cloud-connected pods must have line of sight to the same set of Active Directory domains. If your environment already has cloud-connected Horizon pods and you are deploying your first pod into Microsoft Azure, you must ensure that pod will be able to have line of sight to the Active Directory domains that are already registered with your Horizon Cloud environment. See all of the Active Directory-related topics linked from the Getting Started Using Your Horizon Cloud Environment topic for more details.

DNS Requirements for a Horizon Cloud Pod in Microsoft Azure

For the pod deployment process to deploy your pod successfully into Microsoft Azure, you must configure your firewalls to allow the active pod manager to access the Domain Name Service (DNS) addresses it needs. In addition, your DNS must resolve specific names as described in this topic. In addition to the main pod deployment, when you are deploying the external gateway in
its own VNet, that VNet's subnet must meet the same DNS requirements as the separate pod VNet's management subnet, as described in this topic.

**Important** The pod deployment process uses a jump box VM. This jump box VM has ports and protocol requirements for the pod deployment process, as well as for configuring settings for the pod's Unified Access Gateway VMs when you are deploying a Unified Access Gateway configuration for the pod. See [Ports and Protocols Required by the Pod Jump Box During Pod Deployments and Pod Updates](#).

Deploying the external gateway into its own VNet also uses its own jump box VM, separate from the pod's. That jump box VM has its own ports and protocol requirements for the gateway deployment process. See [When the External Gateway is Deployed in its Own VNet: Ports and Protocols Required by the External Gateway Configuration's Jump Box During Gateway Deployments and Updates](#).

After a pod is successfully deployed, specific ports and protocols are required for ongoing Horizon Cloud operations. The specific ports and protocols required depends on whether the pod is at the manifest version for the September 2019 release, or is at a previous manifest version.

- For a pod created after the September 2019 release or updated to that release's manifest version or later, see [Ports and Protocols Requirements for a Horizon Cloud Pod at the September 2019 Release's Manifest or Later](#). Such pods have manifest versions of 1593 or later.

- For a pod created before the September 2019 release and not yet updated to that release's manifest version, see [Ports and Protocols Requirements for a Horizon Cloud Pod Deployed Prior to the September 2019 Release](#). Such pods have manifest versions of 1493.1 or earlier.

- **DNS Requirements for the Overarching Pod Deployment Process, Pod Updates, and Ongoing Operations**

- **Ports and Protocols Required by the Pod Jump Box During Pod Deployments and Pod Updates**

- **When the External Gateway is Deployed in its Own VNet: Ports and Protocols Required by the External Gateway Configuration's Jump Box During Gateway Deployments and Updates**
DNS Requirements for the Overarching Pod Deployment Process, Pod Updates, and Ongoing Operations

You must ensure the following DNS names are resolvable and reachable from the management and tenant subnets using the specific ports and protocols as indicated in the following table. Horizon Cloud uses specific outbound ports to securely download the pod software into your Microsoft Azure environment and so that the pod can connect back to the Horizon Cloud control plane. You must configure your network firewall, network security group (NSG) rules, and proxy servers such that the active pod manager has the ability to contact the DNS addresses on the ports that it requires. Otherwise, the pod deployment process will fail.

Important

- When you are using the feature to deploy the external gateway into its own VNet, the management subnet in that VNet must meet the same DNS requirements as stated in the table below for the management subnet in the pod’s VNet. The external gateway VNet’s back-end subnet and DMZ subnet do not have specific DNS requirements.

- When you are deploying the pod with either an external gateway, an internal gateway, or both, you must upload a certificate that the pod deployer will configure in those gateway configurations. If the certificate or certificates that you supply for this purpose use CRLs (Certificate Revocation Lists) or OCSP (Online Certificate Status Protocol) settings that refer to specific DNS names, then you must ensure outbound Internet access on the VNet to those DNS names is resolvable and reachable. During configuration of your supplied certificate in the Unified Access Gateway gateway configuration, the Unified Access Gateway software will reach out to those DNS names to check the certificate’s revocation status. If those DNS names are not reachable, pod deployment will fail during its Connecting phase. These names are highly dependent on the CA that you used to obtain the certificates, and therefore are not in VMware’s control.

Your Welcome to Horizon Service email will indicate which regional control plane instance your tenant account was created in. Due to a known issue that existed when the welcome email was sent to you, the email you received might display the system string names used for the regions instead of human-friendly names. If you see a system string name in your welcome email, you can use the following table to relate what is shown in your email with the regional control plane DNS names.

Table 2-11. Regions in Your Welcome Email Mapped to Regional Control Plane DNS Names

<table>
<thead>
<tr>
<th>Your welcome email says</th>
<th>Regional DNS Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>USA</td>
<td>cloud.horizon.vmware.com</td>
</tr>
<tr>
<td>EU_CENTRAL_1 or Europe</td>
<td>cloud-eu-central-1.horizon.vmware.com</td>
</tr>
<tr>
<td>AP_SOUTHEAST_2 or Australia</td>
<td>cloud-op-southeast-2.horizon.vmware.com</td>
</tr>
<tr>
<td>PROD1_NORTHCENTRALUS2_CP1 or USA-2</td>
<td>cloud-us-2.horizon.vmware.com</td>
</tr>
<tr>
<td>PROD1_NORTHEUROPE_CP1 or Europe-2</td>
<td>cloud-eu-2.horizon.vmware.com</td>
</tr>
</tbody>
</table>
### Table 2-11. Regions in Your Welcome Email Mapped to Regional Control Plane DNS Names (continued)

<table>
<thead>
<tr>
<th>Your welcome email says</th>
<th>Regional DNS Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>PROD1_AUSTRALEAST_CP1 or Australia-2</td>
<td>cloud-ap-2.horizon.vmware.com</td>
</tr>
<tr>
<td>Japan</td>
<td>cloud-jp.horizon.vmware.com</td>
</tr>
</tbody>
</table>

### Table 2-12. Pod Deployment and Operations DNS Requirements

<table>
<thead>
<tr>
<th>Subnet Source</th>
<th>Destination (DNS name)</th>
<th>Port</th>
<th>Protocol</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Management</td>
<td>One of the following names, depending on which regional Horizon Cloud control plane instance is specified in your Horizon Cloud tenant account. The regional instance is set when the account is created, as described in Onboarding to Horizon Cloud for Microsoft Azure, Horizon On-Premises, and Horizon on VMware Cloud on AWS.</td>
<td>443</td>
<td>TCP</td>
<td>Regional Horizon Cloud control plane instance</td>
</tr>
<tr>
<td></td>
<td>cloud.horizon.vmware.com</td>
<td></td>
<td></td>
<td>United States: cloud.horizon.vmware.com, cloud-us-2.horizon.vmware.com</td>
</tr>
<tr>
<td></td>
<td>cloud-us-2.horizon.vmware.com</td>
<td></td>
<td></td>
<td>Europe: cloud-eu-central-1.horizon.vmware.com, cloud-eu-2.horizon.vmware.com</td>
</tr>
<tr>
<td></td>
<td>cloud-eu-central-1.horizon.vmware.com</td>
<td></td>
<td></td>
<td>Asia Pacific: cloud-ap-southeast-2.horizon.vmware.com, cloud-ap-2.horizon.vmware.com</td>
</tr>
<tr>
<td></td>
<td>cloud-ap-southeast-2.horizon.vmware.com</td>
<td></td>
<td></td>
<td>Japan: cloud-jp.horizon.vmware.com</td>
</tr>
<tr>
<td></td>
<td>cloud-ap-2.horizon.vmware.com</td>
<td></td>
<td>TCP</td>
<td>VMware software package server. Used for downloading updates of the agent-related software used in the system's image-related operations.</td>
</tr>
<tr>
<td></td>
<td>cloud-jp.horizon.vmware.com</td>
<td></td>
<td>TCP</td>
<td></td>
</tr>
<tr>
<td>Subnet Source</td>
<td>Destination (DNS name)</td>
<td>Port</td>
<td>Protocol</td>
<td>Purpose</td>
</tr>
<tr>
<td>---------------</td>
<td>---------------------------------------------------------------------------------------</td>
<td>------</td>
<td>----------</td>
<td>-------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Management</td>
<td>One of the following names, depending on which of the regional DNS names apply to your account.</td>
<td>443</td>
<td>TCP</td>
<td>Horizon Cloud content delivery server. On the management subnet, this site is used for downloading the VHDs (virtual hard disks) for the pod's manager and Unified Access Gateway VMs. Also used for the VHD for the gateway connector VM, in the case where the external gateway is in its own VNet)</td>
</tr>
<tr>
<td>Subnet Source</td>
<td>Destination (DNS name)</td>
<td>Port</td>
<td>Protocol</td>
<td>Purpose</td>
</tr>
<tr>
<td>---------------</td>
<td>--------------------------------</td>
<td>------------</td>
<td>----------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Management</td>
<td>packages.microsoft.com</td>
<td>443</td>
<td>TCP</td>
<td>Microsoft software package server. Used to securely download the Microsoft Azure Command Line Interface (CLI) software.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>11371</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Management</td>
<td>azure.archive.ubuntu.com</td>
<td>80</td>
<td>TCP</td>
<td>Ubuntu software package server. Used by the pod-related Linux-based VMs for Ubuntu operating system updates.</td>
</tr>
<tr>
<td>Management</td>
<td>api.snapcraft.io</td>
<td>443</td>
<td>TCP</td>
<td>Ubuntu software package server. Used by the pod's Linux-based VMs for Ubuntu operating system updates.</td>
</tr>
<tr>
<td>Management</td>
<td>archive.ubuntu.com</td>
<td>80</td>
<td>TCP</td>
<td>Ubuntu software package server. Used by the pod's Linux-based VMs for Ubuntu operating system updates.</td>
</tr>
<tr>
<td>Management</td>
<td>changelogs.ubuntu.com</td>
<td>80</td>
<td>TCP</td>
<td>Ubuntu software package server. Used by the pod's Linux-based VMs for tracking Ubuntu operating system updates.</td>
</tr>
<tr>
<td>Management</td>
<td>security.ubuntu.com</td>
<td>80</td>
<td>TCP</td>
<td>Ubuntu software package server. Used by the pod's Linux-based VMs for security-related Ubuntu operating system updates.</td>
</tr>
<tr>
<td>Subnet Source</td>
<td>Destination (DNS name)</td>
<td>Port</td>
<td>Protocol</td>
<td>Purpose</td>
</tr>
<tr>
<td>---------------</td>
<td>---------------------------------------------------------------------------------------</td>
<td>------</td>
<td>----------</td>
<td>------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| Management    | One of the following, depending on which Microsoft Azure cloud you are deploying your pod into:  
|               | - Microsoft Azure (global):  
|               | login.microsoftonline.com  
|               | - Microsoft Azure Germany:  
|               | login.microsoftonline.de  
|               | - Microsoft Azure China:  
|               | login.chinacloudapi.cn  
|               | - Microsoft Azure US Government:  
|               | login.microsoftonline.us  
|               | 443                                           | TCP  |  
|               | This web address is generally used by applications to authenticate against Microsoft Azure services. For some descriptions in the Microsoft Azure documentation, see OAuth 2.0 authorization code flow, Azure Active Directory v2.0 and the OpenID Connect protocol, and National clouds. The National clouds topic describes how there are different Azure AD authentication endpoints for each Microsoft Azure national cloud. |  
| Management    | One of the following, depending on which Microsoft Azure cloud you are deploying your pod into:  
|               | - Microsoft Azure (global):  
|               | management.azure.com  
|               | - Microsoft Azure Germany:  
|               | management.microsoftazure.de  
|               | - Microsoft Azure China:  
|               | management.chinacloudapi.cn  
|               | - Microsoft Azure US Government:  
|               | management.usgovcloudapi.net  
|               | 443                                           | TCP  |  
|               | Used for pod API requests to the Microsoft Azure Resource Manager endpoints for using Microsoft Azure Resource Manager services. Microsoft Azure Resource Manager provides a consistent management layer to perform tasks through Azure PowerShell, Azure CLI, Azure portal, REST API, and client SDKs. |
Table 2-12. Pod Deployment and Operations DNS Requirements (continued)

<table>
<thead>
<tr>
<th>Subnet Source</th>
<th>Destination (DNS name)</th>
<th>Port</th>
<th>Protocol</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| Management    | One of the following, depending on which Microsoft Azure cloud you are deploying your pod into:  
- Microsoft Azure (global): graph.windows.net  
- Microsoft Azure Germany: graph.cloudapi.de  
- Microsoft Azure China: graph.chinacloudapi.cn  
- Microsoft Azure US Government: graph.windows.net | 443 | TCP | Access to the Azure Active Directory (Azure AD) Graph API, which is used for the pod’s programmatic access to Azure Active Directory (Azure AD) through OData REST API endpoints. |
| Management    | One of the following, depending on which Microsoft Azure cloud you have deployed your pod into:  
- Microsoft Azure (global): *.blob.core.windows.net  
- Microsoft Azure Germany: *.blob.core.cloudapi.de  
- Microsoft Azure China: *.blob.core.chinacloudapi.cn  
- Microsoft Azure US Government: *.blob.core.usgovcloudapi.net | 443 | TCP | Used for the pod’s programmatic access to the Azure Blob Storage. Azure Blob Storage is a service for storing large amounts of unstructured object data, such as text or binary data. |
| Management    | One of the following, depending on which Microsoft Azure cloud you have deployed your pod into:  
- Microsoft Azure (global): *.vault.azure.net  
- Microsoft Azure Germany: *.vault.microsoftazure.de  
- Microsoft Azure China: *.vault.azure.cn  
- Microsoft Azure US Government: *.vault.usgovcloudapi.net | 443 | TCP | Used for the pod’s ability to programmatically work with the Azure Key Vault cloud service. Azure Key Vault is a cloud service that provides a secure store for secrets. |
Table 2-12. Pod Deployment and Operations DNS Requirements (continued)

<table>
<thead>
<tr>
<th>Subnet Source</th>
<th>Destination (DNS name)</th>
<th>Port</th>
<th>Protocol</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| Management    | If your firewall or network security group (NSG) supports the use of service tags, one of the following:  
- Global Azure SQL service tag: Sql  
- Region-specific SQL service tag for the Azure region where the pod is deployed: Sql.region, such as Sql.WestUS.  
If your firewall or network security group (NSG) does not support the use of service tags, you can use the hostname of the database. This name follows the pattern *.postgres.database.azure.com. | 5432 | TCP      | Used for pod communication to the Microsoft Azure PostgreSQL database server.  
Starting with the September 2019 release, pods that are newly deployed after that release date and pods that are updated to that release’s manifest version are configured with a Microsoft Azure PostgreSQL database server.  
For information about service tags in security groups, see the Microsoft Azure documentation topic at Service tags. |
| Management    | One of the following names, depending on which regional Horizon Cloud control plane instance is specified in your Horizon Cloud tenant account. The regional instance is set when the account is created, as described in Onboarding to Horizon Cloud for Microsoft Azure, Horizon On-Premises, and Horizon on VMware Cloud on AWS.  
- connector-azure-us.vmwarehorizon.com  
- connector-azure-eu.vmwarehorizon.com  
- connector-azure-aus.vmwarehorizon.com  
- connector-azure-jp.vmwarehorizon.com | 443  | TCP      | Regional instance of the Universal Broker service  
- United States: connector-azure-us.vmwarehorizon.com  
- Europe: connector-azure-eu.vmwarehorizon.com  
- Australia: connector-azure-aus.vmwarehorizon.com  
- Japan: connector-azure-jp.vmwarehorizon.com |
<table>
<thead>
<tr>
<th>Subnet Source</th>
<th>Destination (DNS name)</th>
<th>Port</th>
<th>Protocol</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| Tenant        | One of the following names, depending on which of the regional DNS names apply to your account:  
|               |   - d1mes20qfad06k.cloudfront.net  
|               |   - hydra-softwarelib-cdn.azureedge.net                                                 | 443  | TCP      | Horizon Cloud content delivery server. On the tenant subnet, this site is used by the system's automated Import Image process for downloading the installer for the agent-related software.  
|               | d1mes20qfad06k.cloudfront.net corresponds to regional instances for cloud.horizon.vmware.com, cloud-eu-central-1.horizon.vmware.com, cloud-ap-southeast-2.horizon.vmware.com.  
| Tenant        | Depending on which regional Horizon Cloud control plane is specified in your Horizon Cloud account:  
|               |   - kinesis.us-east-1.amazonaws.com  
|               |   - query-prod-us-east-1.cms.vmware.com                                                | 443  | TCP      | Cloud Monitoring Service (CMS)                                         |
|               |   - kinesis.eu-central-1.amazonaws.com                                                |      |          |                                                                         |
|               |   - query-prod-eu-central-1.cms.vmware.com                                             |      |          |                                                                         |
|               |   - kinesis.ap-southeast-2.amazonaws.com                                               |      |          |                                                                         |
Table 2-12. Pod Deployment and Operations DNS Requirements (continued)

<table>
<thead>
<tr>
<th>Subnet Source</th>
<th>Destination (DNS name)</th>
<th>Port</th>
<th>Protocol</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>query-prod-ap-southeast-2.cms.vmware.com</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>kinesis.ap-northeast-1.amazonaws.com</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>query-prod-ap-northeast-1.cms.vmware.com</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Japan:</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Ports and Protocols Required by the Pod Jump Box During Pod Deployments and Pod Updates

As described in Microsoft Azure Virtual Machine Requirements for a Horizon Cloud Pod in Microsoft Azure, a jump box VM is used in the initial creation of a pod and during subsequent software updates on the pod’s environment. After a pod is created, the jump box VM is deleted. Then, when a pod is being updated, the jump box VM is re-created to run that update process and is deleted when the update has completed. Such updates include when a pod is edited to add a Unified Access Gateway configuration.

**Note**  A pod that is either deployed new in Microsoft Azure starting with the September 2019 release or which is updated to the September 2019 release manifest level and has high availability enabled will have two manager VMs. The following paragraphs use the plural word VMs to indicate the jump box VM must communicate with all of the pod’s manager VMs, whether the pod has only one or has two.

During those processes, that jump box VM communicates with:

- The pod’s manager VMs using SSH to the manager VMs’ port 22. As a result, during the pod deployment process and pod update process, the requirement that communication between the jump box VM and the manager VMs' port 22 must be met. The manager VMs' port 22 must be allowed between the jump box VM as a source and the manager VMs as a destination.

- The Unified Access Gateway VMs using HTTPS to those VMs' port 9443, in the case where a pod is deployed with, or edited to add, a Unified Access Gateway configuration. As a result, during the pod deployment process and pod update process, when the pod configuration includes Unified Access Gateway, the requirement that communication between the jump box VM and the Unified Access Gateway VMs' port 9443 must be met. The Unified Access Gateway VMs' port 9443 must be allowed between the jump box VM as a source and the Unified Access Gateway VMs as a destination.

Because these VMs are assigned IP addresses dynamically, the network rules to allow this communication should use:

- The management subnet CIDR as both the source and destination, with destination port 22, source port any, and protocol TCP.
- The management subnet CIDR as both the source and destination, with destination port 9443, source port any, and protocol TCP, when a Unified Access Gateway configuration is involved.

**Note** Ongoing pod operations do not require availability of port 22 on the pod's manager VMs. However, if you make a support request to VMware and the support team determines the way to debug that request is to deploy a jump box VM for SSH communication to your pod's manager VMs, then you will have to meet this port requirement during the time the VMware support team needs the port for debugging your issue. The VMware support team will inform you of any requirements, as appropriate for any support situation.

**When the External Gateway is Deployed in its Own VNet: Ports and Protocols Required by the External Gateway Configuration's Jump Box During Gateway Deployments and Updates**

As described in Microsoft Azure Virtual Machine Requirements for a Horizon Cloud Pod in Microsoft Azure, a jump box VM is used in the initial creation of the external gateway in its own VNet and during subsequent software updates on that gateway. After the external gateway is created in its own VNet, the jump box VM is deleted. Then, when that external gateway is being updated, the jump box VM is re-created to run that update process and is deleted when the update has completed. Such updates include when a pod is edited to add an external gateway in its own VNet.

During those processes, that jump box VM communicates with:

- During those processes, that jump box VM communicates with the gateway connector VM using SSH to that connector VM's port 22. As a result, during the gateway deployment process and update process, the requirement that communication between the jump box VM and the connector VMs' port 22 must be met. The connector VMs' port 22 must be allowed between the jump box VM as a source and the connector VMs as a destination.

- The Unified Access Gateway VMs using HTTPS to those VMs' port 9443. As a result, during the pod deployment process and pod update process, when the pod configuration includes Unified Access Gateway, the requirement that communication between the jump box VM and the Unified Access Gateway VMs' port 9443 must be met. The Unified Access Gateway VMs' port 9443 must be allowed between the jump box VM as a source and the Unified Access Gateway VMs as a destination.

Because these VMs are assigned IP addresses dynamically, the network rules to allow this communication should use:

- The management subnet CIDR as both the source and destination, with destination port 22, source port any, and protocol TCP.
The management subnet CIDR as both the source and destination, with destination port 9443, source port any, and protocol TCP.

**Note** Ongoing pod operations do not require availability of port 22 on the gateway connector's VM. However, if you make a support request to VMware and the support team determines the way to debug that request is to deploy a jump box VM for SSH communication to that gateway's connector VM, then you will have to meet this port requirement during the time the VMware support team needs the port for debugging your issue. The VMware support team will inform you of any requirements, as appropriate for any support situation.

**Ports and Protocols Requirements for a Horizon Cloud Pod at the September 2019 Release's Manifest or Later**

For ongoing Horizon Cloud operations, a pod that is either deployed new in Microsoft Azure starting with the September 2019 release and later, or which is updated to the September 2019 release level, has specific port and protocol requirements that are different from a pod that was deployed previously. Pods deployed new or updated to the September 2019 release have manifest versions of 1600 or later.

**Important** In addition to the ports and protocols described here, you must meet DNS requirements. For details, see [DNS Requirements for a Horizon Cloud Pod in Microsoft Azure](#).

**Ports and Protocols Required by Key Pod Components for Ongoing Operations**

In addition to the DNS requirements, the ports and protocols in the following tables are required for the pod to operate properly for ongoing operations after deployment.

In the tables below, the term manager VM refers to the pod’s manager VM. In the Microsoft Azure portal, this VM has a name that contains a part like `vmw-hcs-podID`, where `podID` is the pod’s UUID, and a node part.

**Important** A pod that is enabled for high availability has two manager VMs. A pod that has high availability disabled has only one manager VM. In the tables below, wherever you see the term manager VM, it applies to all of the manager VMs in your high-availability-enabled pod unless otherwise indicated.

All pods at the September 2019 release's manifest version or later have a pod Microsoft Azure load balancer. The table rows that involve the pod's load balancer apply for all pods at the manifest level of 1600 or later.
<table>
<thead>
<tr>
<th>Source</th>
<th>Target</th>
<th>Ports</th>
<th>Protocol</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Manager VM</td>
<td>Pod's other manager VM</td>
<td>4101</td>
<td>TCP</td>
<td>For a pod that is enabled with high availability, this traffic is JMS routing between the manager VMs.</td>
</tr>
<tr>
<td>Manager VM</td>
<td>Unified Access Gateway VMs</td>
<td>944</td>
<td>HTTPS</td>
<td>This port is used by the pod manager VM over the management subnet to configure settings in the pod’s Unified Access Gateway configuration. This port requirement applies when initially deploying a pod with a Unified Access Gateway configuration and when editing a pod to add a Unified Access Gateway configuration or update settings for that Unified Access Gateway configuration.</td>
</tr>
<tr>
<td>Pod's Microsoft Azure load balancer</td>
<td>Manager VM</td>
<td>808</td>
<td>HTTP</td>
<td>Health checks of the VMs in the load balancer’s backend pool. When a pod at this release’s manifest version is not enabled with high availability, the load balancer has one manager VM is its backend pool to check.</td>
</tr>
<tr>
<td>Manager VM</td>
<td>Domain controller</td>
<td>389</td>
<td>TCP/UDP</td>
<td>LDAP services. Server that contains a domain controller role in an Active Directory configuration. Registering the pod with an Active Directory is a requirement.</td>
</tr>
<tr>
<td>Manager VM</td>
<td>Global catalog</td>
<td>326</td>
<td>TCP</td>
<td>LDAP services. Server that contains global catalog role in an Active Directory configuration. Registering the pod with an Active Directory is a requirement.</td>
</tr>
<tr>
<td>Manager VM</td>
<td>Domain controller</td>
<td>88</td>
<td>TCP/UDP</td>
<td>Kerberos services. Server that contains a domain controller role in an Active Directory configuration. Registering the pod with an Active Directory is a requirement.</td>
</tr>
<tr>
<td>Manager VM</td>
<td>DNS server</td>
<td>53</td>
<td>TCP/UDP</td>
<td>DNS services.</td>
</tr>
<tr>
<td>Manager VM</td>
<td>NTP server</td>
<td>123</td>
<td>UDP</td>
<td>NTP services. Server that provides NTP time synchronization.</td>
</tr>
<tr>
<td>Manager VM</td>
<td>True SSO Enrollment Server</td>
<td>3211</td>
<td>TCP</td>
<td>True SSO Enrollment Server. Optional if you are not using True SSO Enrollment Server capabilities with your pods.</td>
</tr>
<tr>
<td>Manager VM</td>
<td>Workspace ONE Access service</td>
<td>443</td>
<td>HTTPS</td>
<td>Optional if you are not using Workspace ONE Access with the pod. Used to create a trust relationship between the pod and the Workspace ONE Access service. Ensure that the pod can reach the Workspace ONE Access environment you are using, either on-premises or the cloud service, on port 443. If you are using the Workspace ONE Access cloud service, see also the list of Workspace ONE Access service IP addresses to which the Workspace ONE Access Connector and the pod must have access in the VMware Knowledge Base article 2149884.</td>
</tr>
</tbody>
</table>
As described above in *Ports and Protocols Required by the Pod Jump Box During Pod Deployments and Pod Updates*, a transient jump box is used during pod deployment and pod update processes. Even though ongoing processes do not require these ports, during pod deployment and pod update processes, this jump box VM must communicate with the manager VMs using SSH to the manager VMs’ port 22. For details about the cases for which the jump box VM needs this communication, see *Ports and Protocols Required by the Pod Jump Box During Pod Deployments and Pod Updates*.

**Note** A pod that is at manifest version 1600 or later and has the high availability feature enabled on it, will have two manager VMs. The preceding paragraph uses the plural word VMs to indicate the jump box VM must communicate with all of the pod’s manager VMs, whether the pod has only one or has two.

This port is used by the jump box VM over the management subnet to configure settings in the pod’s Unified Access Gateway configuration. This port requirement applies when initially deploying a pod with a Unified Access Gateway configuration and when editing a pod to add a Unified Access Gateway configuration to a pod.

**Gateway Connector VM Ports and Protocols Requirements**

This table applies to the gateway’s connector VM that is used when you have deployed the external gateway in a separate VNet. In addition to the DNS requirements, the ports and protocols in the following table are required for the external gateway to operate properly for ongoing operations after deployment.

In the table below, the term connector VM refers to the gateway’s connector VM which manages the connection between the cloud management plane and the external gateway. In the Microsoft Azure portal, this VM has a name that contains a part like `vmw-hcs-ID`, where `ID` is the gateway’s deployer ID, and a node part.
### Table 2-14. Pod Operations Ports and Protocols

<table>
<thead>
<tr>
<th>Source</th>
<th>Target</th>
<th>Port(s)</th>
<th>Protocol</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connector VM</td>
<td>DNS server</td>
<td>53</td>
<td>TCP, UDP</td>
<td>DNS services.</td>
</tr>
<tr>
<td>NTP server</td>
<td></td>
<td>123</td>
<td>UDP</td>
<td>NTP services. Server that provides NTP time synchronization.</td>
</tr>
<tr>
<td>Transient Jump box VM</td>
<td>Connector VM</td>
<td>22</td>
<td>TCP</td>
<td>As described above in Ports and Protocols Required by the Pod Jump Box During Pod Deployments and Pod Updates, a transient jump box is used during deployment of the external gateway and during update processes. Even though ongoing processes do not require these ports, during deployment and update processes, this jump box VM must communicate with the connector VM using SSH to the connector VMs’ port 22.</td>
</tr>
</tbody>
</table>

### Unified Access Gateway VM Ports and Protocols Requirements

In addition to the DNS and above primary ports and protocols requirements, the ports and protocols in the following tables are related to the gateways that you have configured on the pod to operate properly for ongoing operations after deployment.

For connections using a high-availability-enabled pod configured with Unified Access Gateway instances, traffic must be allowed from the pod’s Unified Access Gateway instances to targets as listed in the table below. During pod deployment, a Network Security Group (NSG) is created in your Microsoft Azure environment for use by the pod’s Unified Access Gateway software.

### Table 2-15. Port Requirements for Traffic from the Pod’s Unified Access Gateway Instances

<table>
<thead>
<tr>
<th>Source</th>
<th>Target</th>
<th>Port</th>
<th>Protocol</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unified Access Gateway</td>
<td>Pod’s Microsoft Azure load balancer</td>
<td>8443</td>
<td>TCP</td>
<td>Login authentication traffic. The traffic from the Unified Access Gateway instances reaches the pod’s manager VM through the pod’s load balancer.</td>
</tr>
<tr>
<td>Unified Access Gateway</td>
<td>Horizon agent in the desktop or farm RDSH VMs</td>
<td>4172</td>
<td>TCP, UDP</td>
<td>PCoIP</td>
</tr>
<tr>
<td>Unified Access Gateway</td>
<td>Horizon agent in the desktop or farm RDSH VMs</td>
<td>22443</td>
<td>TCP, UDP</td>
<td>Blast Extreme By default, when using Blast Extreme, client-drive redirection (CDR) traffic and USB traffic is side-channeled in this port. If you prefer instead, the CDR traffic can be separated onto the TCP 9427 port and the USB redirection traffic can be separated onto the TCP 32111 port.</td>
</tr>
<tr>
<td>Unified Access Gateway</td>
<td>Horizon agent in the desktop or farm RDSH VMs</td>
<td>9427</td>
<td>TCP</td>
<td>Optional for client driver redirection (CDR) and multimedia redirection (MMR) traffic.</td>
</tr>
</tbody>
</table>
### Table 2-15. Port Requirements for Traffic from the Pod's Unified Access Gateway Instances (continued)

<table>
<thead>
<tr>
<th>Source</th>
<th>Target</th>
<th>Port</th>
<th>Protocol</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unified Access Gateway</td>
<td>Horizon agent in the desktop or farm RDSH VMs</td>
<td>32111</td>
<td>TCP</td>
<td>Optional for USB redirection traffic.</td>
</tr>
<tr>
<td>Unified Access Gateway</td>
<td>Your RADIUS instance</td>
<td>1812</td>
<td>UDP</td>
<td>When using RADIUS two-factor authentication with that Unified Access Gateway configuration. The default value for RADIUS is shown here.</td>
</tr>
</tbody>
</table>

### Ports and Protocols Required by Universal Broker

To support the use of Universal Broker for the brokering of end-user assignments from a pod, you must configure port 443 as described in the following table. The active pod manager establishes a persistent WebSocket connection with the Universal Broker service through port 443 and receives connection requests from the Universal Broker service through a randomly selected port.

### Table 2-16. Port Requirements for Universal Broker

<table>
<thead>
<tr>
<th>Source</th>
<th>Source Port</th>
<th>Target</th>
<th>Target Port</th>
<th>Protocol</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Active pod manager</td>
<td>Randomly selected from available ports</td>
<td>Universal Broker service</td>
<td>443</td>
<td>HTTPS initially, then WebSocket</td>
<td>Establishes a persistent WebSocket connection with the Universal Broker service</td>
</tr>
</tbody>
</table>

### End-User Connection Traffic Ports and Protocols Requirements

For detailed information about the various Horizon Clients that your end users might use with your Horizon Cloud pod, see the Horizon Client documentation page at https://docs.vmware.com/en/VMware-Horizon-Client/index.html. Which ports must be opened for traffic from the end users’ connections to reach their pod-provisioned virtual desktops and remote applications depends on the choice you make for how your end users will connect:

**When you choose the deployer option for having an external gateway configuration in the pod’s own VNet**

The deployer deploys Unified Access Gateway instances in your Microsoft Azure environment, along with a Microsoft Azure load balancer resource to those instances in that load balancer’s backend pool. That load balancer communicates with those instances’ NICs.
on the DMZ subnet, and is configured as a **public load balancer in Microsoft Azure**. The diagram **Figure 2-1. Illustration of the Horizon Cloud Pod Architecture for a Pod with High Availability Enabled and Configured with Both External and Internal Gateway Configurations, the External Gateway Deployed into the Same VNet as the Pod, Three NICs on the External Gateway VMs, Two NICs on the Internal Gateway VMs, and a Public IP Enabled for the External Gateway’s Load Balancer** depicts the location of this public load balancer and the Unified Access Gateway instances. When your pod has this configuration, traffic from your end users on the Internet goes to that load balancer, which distributes the requests to the Unified Access Gateway instances. For this configuration, you must ensure that those end-user connections can reach that load balancer using the ports and protocols listed below. Post-deployment, the external gateway's load balancer is located in the resource group named `vmw-hcs-podID-uag`, where `podID` is the pod's UUID.

**When you choose the deployer option for having an internal Unified Access Gateway configuration**

An internal gateway configuration is deployed into the pod's own VNet by default. The deployer deploys Unified Access Gateway instances in your Microsoft Azure environment, along with a **Microsoft Azure load balancer resource** to those instances in its backend pool. That load balancer communicates with those instances' NICs on the tenant subnet, and is configured as an **internal load balancer in Microsoft Azure**. The diagram **Figure 2-1. Illustration of the Horizon Cloud Pod Architecture for a Pod with High Availability Enabled and Configured with Both External and Internal Gateway Configurations, the External Gateway Deployed into the Same VNet as the Pod, Three NICs on the External Gateway VMs, Two NICs on the Internal Gateway VMs, and a Public IP Enabled for the External Gateway’s Load Balancer** depicts the location of this internal load balancer and the Unified Access Gateway instances. When your pod has this configuration, traffic from your end users in your corporate network goes to that load balancer, which distributes the requests to the Unified Access Gateway instances. For this configuration, you must ensure that those end-user connections can reach that load balancer using the ports and protocols listed below. Post-deployment, the internal gateway's load balancer is located in the resource group named `vmw-hcs-podID-uag-internal`, where `podID` is the pod's UUID.

**When you choose the deployer options either for having an external gateway configuration in its own VNet and not the pods, or the option to use its own subscription (which is a special sub-case of using its own VNet because VNets do not span subscriptions)**

The deployer deploys Unified Access Gateway instances in your Microsoft Azure environment, along with a **Microsoft Azure load balancer resource** to those instances in that load balancer's backend pool. That load balancer communicates with those instances' NICs on the DMZ subnet, and is configured as a **public load balancer in Microsoft Azure**. The diagram **Figure 2-2. Illustration of the External Gateway’s Architecture Elements When the External Gateway is Deployed into Its Own VNet, Separate from the Pod’s VNet** depicts the location of this public load balancer and the Unified Access Gateway instances in the gateway's own VNet. When your pod has this configuration, traffic from your end users on the Internet goes to that load balancer, which distributes the requests to the Unified Access Gateway instances.
Gateway instances. For this configuration, you must ensure that those end-user connections can reach that load balancer using the ports and protocols listed below. Post-deployment, the external gateway’s load balancer is located in the resource group named `vmw-hcs-\(ID\)-uag`, where \(ID\) is the value shown in the **Deployer ID** field of the pod’s details page. As described in the *Administration Guide*, you get to the pod’s details page from the console’s Capacity page.

**When you choose to have zero Unified Access Gateway configurations on the pod, such as when you are integrating Workspace ONE Access with the pod or are allowing internal users to connect directly over VPN**

**Attention** In production systems, for internal-user access, the best practice is to use an internal Unified Access Gateway gateway configuration on the pod, and not direct connections to the pod.

When Workspace ONE Access is integrated with the pod, you typically have your end users connect through Workspace ONE Access. When Workspace ONE Access is integrated with a Horizon Cloud pod in Microsoft Azure, you must configure it pointing directly to the pod. No Unified Access Gateway configuration is needed on the pod then, when your end users are connecting to their pod-provisioned resources using Workspace ONE Access. For this configuration, you upload an SSL certificate to the pod’s manager VMs using the pod’s summary page in the console, as described in the *VMware Horizon Cloud Service Administration Guide*. Then you complete the steps to integrate Workspace ONE Access with the pod.

**Table 2-17. External End User Connections Ports and Protocols when the Pod Configuration has External Unified Access Gateway instances**

<table>
<thead>
<tr>
<th>Source</th>
<th>Target</th>
<th>Port</th>
<th>Protocol</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Horizon Client</td>
<td>Microsoft Azure load balancer for these Unified Access Gateway instances</td>
<td>443</td>
<td>TCP</td>
<td>Login authentication traffic. Can also carry client-drive redirection (CDR), multimedia redirection (MMR), USB redirection, and tunneled RDP traffic. SSL (HTTPS access) is enabled by default for client connections. Port 80 (HTTP access) can be used in some cases. See the topic Understanding What URL Content Redirection Is in the <em>VMware Horizon Cloud Service Administration Guide</em>.</td>
</tr>
<tr>
<td>Horizon Client</td>
<td>Microsoft Azure load balancer for these Unified Access Gateway instances</td>
<td>4172</td>
<td>TCP/UDP</td>
<td>PCoIP via PCoIP Secure Gateway on Unified Access Gateway</td>
</tr>
</tbody>
</table>
### Table 2-17. External End User Connections Ports and Protocols when the Pod Configuration has External Unified Access Gateway instances (continued)

<table>
<thead>
<tr>
<th>Source</th>
<th>Target</th>
<th>Port</th>
<th>Protocol</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Horizon Client</td>
<td>Microsoft Azure load balancer for these Unified Access Gateway instances</td>
<td>443</td>
<td>TCP</td>
<td>Blast Extreme via Blast Secure Gateway on Unified Access Gateway for data traffic.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Microsoft Azure load balancer for these Unified Access Gateway instances</td>
<td>443</td>
<td>UDP</td>
<td>Blast Extreme via the Unified Access Gateway for data traffic.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Microsoft Azure load balancer for these Unified Access Gateway instances</td>
<td>844</td>
<td>UDP</td>
<td>Blast Extreme via Blast Secure Gateway on Unified Access Gateway for data traffic (adaptive transport).</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Browser</td>
<td>Microsoft Azure load balancer for these Unified Access Gateway instances</td>
<td>443</td>
<td>TCP</td>
<td>HTML Access</td>
</tr>
</tbody>
</table>

### Table 2-18. Internal End User Connections Ports and Protocols when the Pod Configuration has Internal Unified Access Gateway instances

<table>
<thead>
<tr>
<th>Source</th>
<th>Target</th>
<th>Port</th>
<th>Protocol</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Horizon Client</td>
<td>Microsoft Azure load balancer for these Unified Access Gateway instances</td>
<td>443</td>
<td>TCP</td>
<td>Login authentication traffic. Can also carry client-drive redirection (CDR), multimedia redirection (MMR), USB redirection, and tunneled RDP traffic. SSL (HTTPS access) is enabled by default for client connections. Port 80 (HTTP access) can be used in some cases. See the topic Understanding What URL Content Redirection Is in the VMware Horizon Cloud Service Administration Guide.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Horizon Client</td>
<td>Microsoft Azure load balancer for these Unified Access Gateway instances</td>
<td>4172</td>
<td>TCP/UDP</td>
<td>PCoIP via PCoIP Secure Gateway on Unified Access Gateway</td>
</tr>
</tbody>
</table>
### Table 2-18. Internal End User Connections Ports and Protocols when the Pod Configuration has Internal Unified Access Gateway instances (continued)

<table>
<thead>
<tr>
<th>Source</th>
<th>Target</th>
<th>Port</th>
<th>Protocol</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Horizon Client</td>
<td>Microsoft Azure load balancer for these Unified Access Gateway instances</td>
<td>443</td>
<td>TCP</td>
<td>Blast Extreme via Blast Secure Gateway on Unified Access Gateway for data traffic.</td>
</tr>
<tr>
<td>Horizon Client</td>
<td>Microsoft Azure load balancer for these Unified Access Gateway instances</td>
<td>443</td>
<td>UDP</td>
<td>Blast Extreme via the Unified Access Gateway for data traffic.</td>
</tr>
<tr>
<td>Horizon Client</td>
<td>Microsoft Azure load balancer for these Unified Access Gateway instances</td>
<td>844</td>
<td>UDP</td>
<td>Blast Extreme via Blast Secure Gateway on Unified Access Gateway for data traffic (adaptive transport).</td>
</tr>
<tr>
<td>Browser</td>
<td>Microsoft Azure load balancer for these Unified Access Gateway instances</td>
<td>443</td>
<td>TCP</td>
<td>HTML Access</td>
</tr>
</tbody>
</table>

### Table 2-19. Internal End User Connections Ports and Protocols when using Direct Pod Connections, Such as Over VPN

<table>
<thead>
<tr>
<th>Source</th>
<th>Target</th>
<th>Port</th>
<th>Protocol</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Horizon Client</td>
<td>Pod's Microsoft Azure load balancer</td>
<td>443</td>
<td>TCP</td>
<td>Login authentication traffic. The traffic from the clients reaches the pod’s manager VMs through the pod's load balancer.</td>
</tr>
<tr>
<td>Horizon Client</td>
<td>Horizon agent in the desktop or farm RDSH VMs</td>
<td>4172</td>
<td>TCP</td>
<td>PCoIP</td>
</tr>
<tr>
<td>Horizon Client</td>
<td>Horizon agent in the desktop or farm RDSH VMs</td>
<td>22443</td>
<td>TCP</td>
<td>Blast Extreme</td>
</tr>
</tbody>
</table>
### Table 2-19. Internal End User Connections Ports and Protocols when using Direct Pod Connections, Such as Over VPN (continued)

<table>
<thead>
<tr>
<th>Source</th>
<th>Target</th>
<th>Port</th>
<th>Protocol</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Horizon Client</td>
<td>Horizon agent in the desktop or farm RDSH VMs</td>
<td>32111</td>
<td>TCP</td>
<td>USB redirection</td>
</tr>
<tr>
<td>Horizon Client</td>
<td>Horizon agent in the desktop or farm RDSH VMs</td>
<td>9427</td>
<td>TCP</td>
<td>Client-drive redirection (CDR) and multimedia redirection (MMR)</td>
</tr>
<tr>
<td>Browser</td>
<td>Horizon agent in the desktop or farm RDSH VMs</td>
<td>443</td>
<td>TCP</td>
<td>HTML Access</td>
</tr>
</tbody>
</table>

**Ports and Protocols Requirements for Traffic from the Horizon Agent in the Base VM, VDI Desktop VMs, and Farm RDSH VMs**

The following ports must allow traffic between the Horizon agent-related software that is installed in the base VMs, desktop VMs, and farm RDSH VMs and the pod's manager VMs.
<table>
<thead>
<tr>
<th>Source</th>
<th>Target</th>
<th>Port</th>
<th>Protocol</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Horizon agent in the base imported VM, golden images, desktop VMs, farm RDSH VMs</td>
<td>Manager VM</td>
<td>4001</td>
<td>TCP</td>
<td>Java Message Service (JMS, non-SSL), used by the agent in the VM to communicate with the pod as part of the certificate thumbprint verification and exchange to secure an SSL connection to the pod. After the keys are negotiated and exchanged between the VM and the pod manager, the agent uses port 4002 to create a secured SSL connection. For example, the Reset Agent Pairing action on the Imported VMs page requires communication over port 4001 for that agent pairing workflow between the base imported VM and the pod. <strong>Note</strong> Both ports 4001 and 4002 are required for steady-state operations. There are times when the agent might need to re-key with the pod, so port 4001 must be kept open.</td>
</tr>
<tr>
<td>Horizon agent in the base imported VM, the golden images, desktop VMs, farm RDSH VMs</td>
<td>Manager VM</td>
<td>4002</td>
<td>TCP</td>
<td>Java Message Service (JMS, SSL), used by the agent in these VMs to communicate with the pod using a secured SSL connection.</td>
</tr>
<tr>
<td>FlexEngine agent (the agent for VMware Dynamic Environment Manager) in the desktop or farm RDSH VMs</td>
<td>Those file shares that you set up for use by the FlexEngine agent that runs in the desktop or farm RDSH VMs</td>
<td>445</td>
<td>TCP</td>
<td>FlexEngine agent access to your SMB file shares, if you are using VMware Dynamic Environment Manager capabilities.</td>
</tr>
</tbody>
</table>

As part of the pod deployment process, the deployer creates network security groups (NSGs) on the network interfaces (NICs) on all of the deployed VMs. For details about the rules defined in those NSGs, see the **Horizon Cloud Administration Guide**.

**Note** Instead of listing DNS names, IP addresses, ports, and protocols in a Horizon Cloud Knowledge Base (KB) article, we have provided them here as part of the core Horizon Cloud documentation.

**Ports and Protocols Requirements for a Horizon Cloud Pod Deployed Prior to the September 2019 Release**

For ongoing Horizon Cloud operations, a pod that was deployed in Microsoft Azure prior to the September 2019 release has specific port and protocol requirements that are different from a
A pod that is deployed at the manifest version of the September 2019 release, or which is updated to the September 2019 release's manifest version. A pod that was deployed prior to the September 2019 release has a manifest version of 1493.1 or earlier.

**Important** In addition to the ports and protocols described here, you must meet DNS requirements. For details, see [DNS Requirements for a Horizon Cloud Pod in Microsoft Azure](#).

### Ports and Protocols Required for Ongoing Operations for a Pod of Manifest Version

In addition to the DNS requirements, the ports and protocols in the following tables are required for the pod to operate properly for ongoing operations after deployment.

**Note** In this section's tables, the term manager VM refers to the pod's manager VM. In the Microsoft Azure portal, this VM has a name that contains a part like `vmw-hcs-podID`, where `podID` is the pod's UUID, and a `node` part.

#### Table 2-20. Pod Operations Ports and Protocols

<table>
<thead>
<tr>
<th>Source</th>
<th>Target</th>
<th>Ports</th>
<th>Protocol</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Manager VM</td>
<td>Domain controller</td>
<td>389</td>
<td>TCP, UDP</td>
<td>LDAP services. Server that contains a domain controller role in an Active Directory configuration. Registering the pod with an Active Directory is a requirement.</td>
</tr>
<tr>
<td>Manager VM</td>
<td>Global catalog</td>
<td>326</td>
<td>TCP</td>
<td>LDAP services. Server that contains global catalog role in an Active Directory configuration. Registering the pod with an Active Directory is a requirement.</td>
</tr>
<tr>
<td>Manager VM</td>
<td>Domain controller</td>
<td>88</td>
<td>TCP, UDP</td>
<td>Kerberos services. Server that contains a domain controller role in an Active Directory configuration. Registering the pod with an Active Directory is a requirement.</td>
</tr>
<tr>
<td>Manager VM</td>
<td>DNS server</td>
<td>53</td>
<td>TCP, UDP</td>
<td>DNS services.</td>
</tr>
<tr>
<td>Manager VM</td>
<td>NTP server</td>
<td>123</td>
<td>UDP</td>
<td>NTP services. Server that provides NTP time synchronization.</td>
</tr>
<tr>
<td>Manager VM</td>
<td>True SSO Enrollment Server</td>
<td>3211</td>
<td>TCP</td>
<td>True SSO Enrollment Server. Optional if you are not using True SSO Enrollment Server capabilities with your pods.</td>
</tr>
</tbody>
</table>
Which ports must be opened for traffic from the end users' connections to reach their pod-provisioned virtual desktops and remote applications depends on the choice you make for how your end users will connect:

- When you choose the option for having an external gateway configuration, Unified Access Gateway instances are automatically deployed in your Microsoft Azure environment, along with a Microsoft Azure load balancer resource to those instances in its backend pool. That load balancer communicates with those instances' NICs on the DMZ subnet, and is configured as a public load balancer in Microsoft Azure. The diagram Figure 2-1. Illustration of the Horizon Cloud Pod Architecture for a Pod with High Availability Enabled and Configured with Both External and Internal Gateway Configurations, the External Gateway Deployed into the Same VNet as the Pod, Three NICs on the External Gateway VMs, Two NICs on the Internal Gateway VMs, and a Public IP Enabled for the External Gateway’s Load Balancer depicts the location of this public load balancer and the Unified Access Gateway instances. When your pod has this configuration, traffic from your end users on the Internet goes to that load balancer, which distributes the requests to the Unified Access Gateway instances. For this configuration, you must ensure that those end-user connections can reach that load balancer using the ports and protocols listed below. For the deployed pod, the external gateway’s load balancer is located in the resource group named `vmw-hcs-podID-uag`, where `podID` is the pod’s UUID.

- When you choose the option for having an internal Unified Access Gateway configuration, Unified Access Gateway instances are automatically deployed in your Microsoft Azure environment, along with a Microsoft Azure load balancer resource to those instances in its backend pool. That load balancer communicates with those instances' NICs on the tenant subnet, and is configured as an internal load balancer in Microsoft Azure. The diagram Figure
2-1. Illustration of the Horizon Cloud Pod Architecture for a Pod with High Availability Enabled and Configured with Both External and Internal Gateway Configurations, the External Gateway Deployed into the Same VNet as the Pod, Three NICs on the External Gateway VMs, Two NICs on the Internal Gateway VMs, and a Public IP Enabled for the External Gateway's Load Balancer depicts the location of this internal load balancer and the Unified Access Gateway instances. When your pod has this configuration, traffic from your end users in your corporate network goes to that load balancer, which distributes the requests to the Unified Access Gateway instances. For this configuration, you must ensure that those end-user connections can reach that load balancer using the ports and protocols listed below. For the deployed pod, the internal gateway's load balancer is located in the resource group named `vmw-hcs-podID-uag-internal`, where `podID` is the pod's UUID.

- When you do not choose either Unified Access Gateway configurations, you can instead have your end users connecting directly to the pod, such as using a VPN. For this configuration, you upload an SSL certificate to the pod's manager VM using the pod's summary page in the administrative console, as described in the VMware Horizon Cloud Service Administration Guide.

For detailed information about the various Horizon Clients that your end users might use with your Horizon Cloud pod, see the Horizon Client documentation page at https://docs.vmware.com/en/VMware-Horizon-Client/index.html.

**Table 2-21. External End User Connections Ports and Protocols when the Pod Configuration has External Unified Access Gateway instances**

<table>
<thead>
<tr>
<th>Source</th>
<th>Target</th>
<th>Port</th>
<th>Protocol</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Horizon Client</td>
<td>Microsoft Azure load balancer for these Unified Access Gateway instances</td>
<td>443</td>
<td>TCP</td>
<td>Login authentication traffic. Can also carry client-drive redirection (CDR), multimedia redirection (MMR), USB redirection, and tunneled RDP traffic. SSL (HTTPS access) is enabled by default for client connections. Port 80 (HTTP access) can be used in some cases. See the topic Understanding What URL Content Redirection Is in the VMware Horizon Cloud Service Administration Guide.</td>
</tr>
<tr>
<td>Horizon Client</td>
<td>Microsoft Azure load balancer for these Unified Access Gateway instances</td>
<td>4172</td>
<td>TCP, UDP</td>
<td>PCoIP via PCoIP Secure Gateway on Unified Access Gateway</td>
</tr>
<tr>
<td>Horizon Client</td>
<td>Microsoft Azure load balancer for these Unified Access Gateway instances</td>
<td>443</td>
<td>TCP</td>
<td>Blast Extreme via Blast Secure Gateway on Unified Access Gateway for data traffic.</td>
</tr>
</tbody>
</table>
### Table 2-21. External End User Connections Ports and Protocols when the Pod Configuration has External Unified Access Gateway instances (continued)

<table>
<thead>
<tr>
<th>Source</th>
<th>Target</th>
<th>Port</th>
<th>Protocol</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Horizon Client</td>
<td>Microsoft Azure load balancer for these Unified Access Gateway instances</td>
<td>443</td>
<td>UDP</td>
<td>Blast Extreme via the Unified Access Gateway for data traffic.</td>
</tr>
<tr>
<td>Horizon Client</td>
<td>Microsoft Azure load balancer for these Unified Access Gateway instances</td>
<td>844</td>
<td>UDP</td>
<td>Blast Extreme via Blast Secure Gateway on Unified Access Gateway for data traffic (adaptive transport).</td>
</tr>
<tr>
<td>Browser</td>
<td>Microsoft Azure load balancer for these Unified Access Gateway instances</td>
<td>443</td>
<td>TCP</td>
<td>HTML Access</td>
</tr>
</tbody>
</table>

### Table 2-22. Internal End User Connections Ports and Protocols when the Pod Configuration has Internal Unified Access Gateway instances

<table>
<thead>
<tr>
<th>Source</th>
<th>Target</th>
<th>Port</th>
<th>Protocol</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Horizon Client</td>
<td>Microsoft Azure load balancer for these Unified Access Gateway instances</td>
<td>443</td>
<td>TCP</td>
<td>Login authentication traffic. Can also carry client-drive redirection (CDR), multimedia redirection (MMR), USB redirection, and tunneled RDP traffic. SSL (HTTPS access) is enabled by default for client connections. Port 80 (HTTP access) can be used in some cases. See the topic Understanding What URL Content Redirection Is in the VMware Horizon Cloud Service Administration Guide.</td>
</tr>
<tr>
<td>Horizon Client</td>
<td>Microsoft Azure load balancer for these Unified Access Gateway instances</td>
<td>4172</td>
<td>TCP</td>
<td>PCoIP via PCoIP Secure Gateway on Unified Access Gateway</td>
</tr>
<tr>
<td>Horizon Client</td>
<td>Microsoft Azure load balancer for these Unified Access Gateway instances</td>
<td>443</td>
<td>TCP</td>
<td>Blast Extreme via Blast Secure Gateway on Unified Access Gateway for data traffic.</td>
</tr>
</tbody>
</table>
Table 2-22. Internal End User Connections Ports and Protocols when the Pod Configuration has Internal Unified Access Gateway instances (continued)

<table>
<thead>
<tr>
<th>Source</th>
<th>Target</th>
<th>Port</th>
<th>Protocol</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Horizon Client</td>
<td>Microsoft Azure load balancer for these Unified Access Gateway instances</td>
<td>443</td>
<td>UDP</td>
<td>Blast Extreme via the Unified Access Gateway for data traffic.</td>
</tr>
<tr>
<td>Horizon Client</td>
<td>Microsoft Azure load balancer for these Unified Access Gateway instances</td>
<td>844</td>
<td>UDP</td>
<td>Blast Extreme via Blast Secure Gateway on Unified Access Gateway for data traffic (adaptive transport).</td>
</tr>
<tr>
<td>Browser</td>
<td>Microsoft Azure load balancer for these Unified Access Gateway instances</td>
<td>443</td>
<td>TCP</td>
<td>HTML Access</td>
</tr>
</tbody>
</table>

Table 2-23. Internal End User Connections Ports and Protocols when using Direct Pod Connections, Such as Over VPN

<table>
<thead>
<tr>
<th>Source</th>
<th>Target</th>
<th>Port</th>
<th>Protocol</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Horizon Client</td>
<td>Manager VM</td>
<td>443</td>
<td>TCP</td>
<td>Login authentication traffic</td>
</tr>
<tr>
<td>Horizon Client</td>
<td>Horizon agent in the desktop or farm server VMs</td>
<td>4172</td>
<td>TCP, UDP</td>
<td>PCoIP</td>
</tr>
<tr>
<td>Horizon Client</td>
<td>Horizon agent in the desktop or farm server VMs</td>
<td>22443</td>
<td>TCP, UDP</td>
<td>Blast Extreme</td>
</tr>
<tr>
<td>Horizon Client</td>
<td>Horizon agent in the desktop or farm server VMs</td>
<td>32111</td>
<td>TCP</td>
<td>USB redirection</td>
</tr>
</tbody>
</table>
Table 2-23. Internal End User Connections Ports and Protocols when using Direct Pod Connections, Such as Over VPN (continued)

<table>
<thead>
<tr>
<th>Source</th>
<th>Target</th>
<th>Port</th>
<th>Protocol</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Horizon Client</td>
<td>Horizon agent in the desktop or farm server VMs</td>
<td>9427</td>
<td>TCP</td>
<td>Client-drive redirection (CDR) and multimedia redirection (MMR)</td>
</tr>
<tr>
<td>Browser</td>
<td>Horizon agent in the desktop or farm server VMs</td>
<td>443</td>
<td>TCP</td>
<td>HTML Access</td>
</tr>
</tbody>
</table>

For connections using a pod configured with Unified Access Gateway instances, traffic must be allowed from the pod’s Unified Access Gateway instances to targets as listed in the table below. During pod deployment, a Network Security Group (NSG) is created in your Microsoft Azure environment for use by the pod’s Unified Access Gateway software.

Table 2-24. Port Requirements for Traffic from the Pod’s Unified Access Gateway Instances

<table>
<thead>
<tr>
<th>Source</th>
<th>Target</th>
<th>Port</th>
<th>Protocol</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unified Access Gateway</td>
<td>Manager VM</td>
<td>443</td>
<td>TCP</td>
<td>Login authentication traffic</td>
</tr>
<tr>
<td>Unified Access Gateway</td>
<td>Horizon agent in the desktop or farm server VMs</td>
<td>4172</td>
<td>TCP, UDP</td>
<td>PCoIP</td>
</tr>
<tr>
<td>Unified Access Gateway</td>
<td>Horizon agent in the desktop or farm server VMs</td>
<td>22443</td>
<td>TCP, UDP</td>
<td>Blast Extreme By default, when using Blast Extreme, client-drive redirection (CDR) traffic and USB traffic is side-channeled in this port. If you prefer instead, the CDR traffic can be separated onto the TCP 9427 port and the USB redirection traffic can be separated onto the TCP 32111 port.</td>
</tr>
<tr>
<td>Unified Access Gateway</td>
<td>Horizon agent in the desktop or farm server VMs</td>
<td>9427</td>
<td>TCP</td>
<td>Optional for client driver redirection (CDR) and multimedia redirection (MMR) traffic.</td>
</tr>
<tr>
<td>Unified Access Gateway</td>
<td>Horizon agent in the desktop or farm server VMs</td>
<td>32111</td>
<td>TCP</td>
<td>Optional for USB redirection traffic.</td>
</tr>
<tr>
<td>Unified Access Gateway</td>
<td>Your RADIUS instance</td>
<td>1812</td>
<td>UDP</td>
<td>When using RADIUS two-factor authentication with that Unified Access Gateway configuration. The default value for RADIUS is shown here.</td>
</tr>
</tbody>
</table>

The following ports must allow traffic from the Horizon agent-related software that is installed in the desktop VMs and farm server VMs.
<table>
<thead>
<tr>
<th>Source</th>
<th>Target</th>
<th>Port</th>
<th>Protocol</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Horizon agent in the desktop or farm server VMs</td>
<td>Manager VM</td>
<td>4001</td>
<td>TCP</td>
<td>Java Message Service (JMS, non-SSL), used by the agent in the VM when the agent is not yet paired with the pod. The agent communicates with the pod to get the information it needs to pair with the pod. After the agent is paired, it uses port 4002 to communicate with the pod.</td>
</tr>
<tr>
<td>Horizon agent in the desktop or farm server VMs</td>
<td>Manager VM</td>
<td>4002</td>
<td>TCP</td>
<td>Java Message Service (JMS, SSL), used by the agent to communicate with the pod when the agent is already paired with the pod.</td>
</tr>
<tr>
<td>FlexEngine agent (the agent for VMware Dynamic Environment Manager) in the desktop or farm server VMs</td>
<td>Those file shares that you set up for use by the FlexEngine agent that runs in the desktop or farm server VMs</td>
<td>445</td>
<td>TCP</td>
<td>FlexEngine agent access to your SMB file shares, if you are using VMware Dynamic Environment Manager capabilities.</td>
</tr>
</tbody>
</table>

As part of the pod deployment process, the deployer creates network security groups (NSGs) on the network interfaces (NICs) on all of the deployed VMs. For details about the rules defined in those NSGs, see the Horizon Cloud Administration Guide.

**Note** Instead of listing DNS names, IP addresses, ports, and protocols in a Horizon Cloud Knowledge Base (KB) article, we have provided them here as part of the core Horizon Cloud documentation.

**Create the Required Service Principal Needed by the Horizon Cloud Pod Deployer by Creating an Application Registration**

The Horizon Cloud pod deployer needs a service principal to access and use your Microsoft Azure subscription's capacity for your Horizon Cloud pods. When you register a Microsoft Azure AD application, the service principal is also created. Also, you must generate an authentication key and assign a role to the service principal at the subscription level. If you are going to use the feature to have the external gateway use its own subscription, separate from the pod's, you must also perform similar steps for a service principal associated with that subscription.
For up-to-date and in-depth details and screenshots for creating a service principal, see the Microsoft Azure documentation's documentation topic How to: Use the portal to create an Azure AD application and service principal that can access resources.

**Important** Each service principal that you configure for Horizon Cloud's use must be assigned an appropriate role in that service principal's associated subscription. The role to a service principal must allow the actions that Horizon Cloud needs to operated on the Horizon Cloud managed resources in that service principal's associated Microsoft Azure subscription. The service principal for the pod's subscription needs a role that allows for actions to successfully deploy the pod, to operate on the pod and the pod-managed resources to fulfill the administrator workflows initiated using the administrative console, and to maintain the pod over time. When using a separate subscription for the pod's external Unified Access Gateway configuration, the service principal for that subscription needs a role that allows for actions to successfully deploy the resources needed for that gateway configuration, to operate on those Horizon Cloud-managed resources to fulfill the administrator workflows, and to maintain those gateway-related resources over time.

As described in *Operations Required by Horizon Cloud in Your Microsoft Azure Subscriptions*, the service principal must be granted access using one of the following methods:

- At the subscription level, assign the Contributor role. The Contributor role is one of the Microsoft Azure built-in roles. The Contributor role is described in *Built-in roles for Azure resources* in the Microsoft Azure documentation.

- At the subscription level, assign a custom role that you have set up to provide the service principal with *Operations Required by Horizon Cloud in Your Microsoft Azure Subscriptions* that Horizon Cloud needs for deployment of the pod-related resources and for ongoing administrator-initiated workflows and pod maintenance operations.

- When using a separate subscription for the external Unified Access Gateway configuration and deploying into an existing resource group, an valid combination is to grant access to the service principal to access that resource group and associated VNet using a role that provides narrow-scope permissions plus grant access for the service principal to access the subscription using the built-in Reader role.

You perform these steps using the Microsoft Azure portal appropriate for your registered account. For example, there are specific portal endpoints for these Microsoft Azure clouds.

- Microsoft Azure (standard global)
- Microsoft Azure China
Microsoft Azure US Government

**Note** When performing these steps, you can collect some of the values that you will need for the deployment wizard, as described in *Subscription-Related Information for the Horizon Cloud Pod Deployment Wizard*, specifically:

- Application ID
- Authentication key

**Caution** Even though you can set the secret key’s expiration duration to a specific timeframe, if you do that, you must remember to refresh the key before it expires or the associated Horizon Cloud pod will stop working. Horizon Cloud cannot detect or know what duration you set. For smooth operations, set the key’s expiration to **Never**.

If you prefer not to set the expiration to **Never** and prefer instead to refresh the key before it expires, you must remember to log in to Horizon Cloud before the expiration date, and in the pod’s details where the subscription information is listed, enter the new key value in the associated pod’s subscription information. For detailed steps, see the *Update the Subscription Information Associated with Deployed Pods* topic in the *Administration Guide*.

In the steps below, Step 7.a illustrates the service principal being granted access at the subscription level.

**Prerequisites**

If you want to assign a custom role to the service principal instead of the built-in **Contributor** role, verify that the custom role exists in your subscription. Verify that the custom role permits the management operations required by Horizon Cloud, as described in *Operations Required by Horizon Cloud in Your Microsoft Azure Subscriptions*.

**Procedure**

1. From the Microsoft Azure portal’s left navigation bar, click **Azure Active Directory** (Azure Active Directory), then click **App registrations** (App registrations).
2. Click **New application registration**.
3. Type a descriptive name and select a supported account type.
4. In the Redirect URI section, select **Web**, type `http://localhost:8000`, and click **Register**.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Name</strong></td>
<td>The name is up to you. The name is a way you can differentiate this service principal used by Horizon Cloud from any other service principals that might exist in this same subscription.</td>
</tr>
<tr>
<td><strong>Redirect URI</strong></td>
<td>Ensure <strong>Web</strong> is selected. Type <code>http://localhost:8000</code> as shown. Microsoft Azure marks this as a required field. Because Horizon Cloud does not need a sign-on URL for the service principal <code>http://localhost:8000</code> is used to satisfy the Microsoft Azure requirement.</td>
</tr>
</tbody>
</table>

The newly created app registration is displayed on screen.

5. Copy the application ID and directory (tenant) ID and save them to a location where you can retrieve them later when you run the deployment wizard.

6. From the service principal's details screen, create the service principal's authentication secret key.
   a. Click **Certificates & secrets** (Certificates and secrets).
   b. Click **New client secret**.
c Type a description, select an expiration duration, and click **Add**.

The key description must be 16 characters or less, for example *Hzn-Cloud-Key1*.

**Caution** You can set the expiration duration to **Never** or to a specific timeframe. However, if you set a specific duration, you must remember to refresh the key before it expires and enter the new key into Horizon Cloud, using the console and in the pod's details where the pod's subscription information is listed. Otherwise, the associated pod will stop working. Horizon Cloud cannot detect or know what duration you set in the Microsoft Azure portal.

---

**Important** Keep this screen open until you copy the secret value and paste the value into a location where you can retrieve it later. Do not close the screen until you have copied the secret value.

<table>
<thead>
<tr>
<th>DESCRIPTION</th>
<th>EXPIRES</th>
<th>VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hzn-cloud-key1</td>
<td>12/31/2299</td>
<td></td>
</tr>
</tbody>
</table>

d Copy the secret value to a location where you can retrieve it later when you run the deployment wizard.
7 Assign a role to the service principal at the subscription level.

**Caution** If the service principal's assigned role does not permit the operations that the pod deployer requires, according to the options you select in the deployment wizard, the wizard will block you from completing the wizard's steps. For the permissions the assigned role must provide, see *Operations Required by Horizon Cloud in Your Microsoft Azure Subscriptions*.

a Navigate to your subscription's settings screen by clicking **All services** in the Microsoft Azure portal's main navigation bar, clicking **Subscriptions**, and then clicking the name of the subscription that you will use with the pod.

**Note** At this point, from the screen, you can copy the subscription ID which you will later need in the deployment wizard.

b Click **Access control (IAM)** (Access control (IAM)) and then click **Add > Add role assignment** to open the **Add role assignment** screen.

c In the **Add role assignment** screen, for **Role**, select the role you are assigning, according to the rules described in *Operations Required by Horizon Cloud in Your Microsoft Azure Subscriptions*. 
d  Use the **Select** box to search for your service principal by the name you gave it.

The following screenshot illustrates this step where the Contributor role is selected for the service principal.

![Role Selection Screenshot]

**Note**  Make sure the **Assign access to** drop-down list is set to **Azure AD user, group, or application**.

e  Click your service principal to make it a selected member and then click **Save**.
Verify that your subscription has the registered resource providers that the pod requires.

a. From the Access control (IAM) screen you are on from the previous step, navigate to the subscription's list of resource providers by clicking (Resource providers) in the subscription's menu.

b. Verify that the following resource providers have (Registered) status, and if not, register them.

- Microsoft.Compute
- microsoft.insights
- Microsoft.Network
- Microsoft.Storage
- Microsoft.KeyVault
- Microsoft.Authorization
- Microsoft.Resources
- Microsoft.ResourceHealth
- Microsoft.DBforPostgreSQL
- Microsoft.Sql

Results

At this point, you've created and configured the service provider for the pod, and you have the subscription-related values you need in the first step of the pod deployment wizard. The four subscription-related values are:

- Subscription ID
- Azure Active Directory ID
- Application ID
- Application key value

**What to do next**

Verify that you have collected all of the subscription-related information you will enter in the deployment wizard. See Subscription-Related Information for the Horizon Cloud Pod Deployment Wizard.

If you are going to use a separate subscription for deploying the external Unified Access Gateway configuration into an existing resource group, and you want to grant granular, narrow-scope permissions instead of access at the subscription level, see Operations Required by Horizon Cloud in Your Microsoft Azure Subscriptions for details. Ensure the appropriate access is granted to the service principal to meet the Horizon Cloud deployer's requirements.

**Operations Required by Horizon Cloud in Your Microsoft Azure Subscriptions**

The service principal used by Horizon Cloud to perform operations in your Microsoft Azure subscription and resource groups needs an assigned role that specifies the permitted operations that service principal can perform in that subscription and its resource groups. Even though using the Microsoft Azure built-in Contributor role provides for all of the operations needed by Horizon Cloud, it does that by granting the broadest number of permissions. Instead of using that Microsoft Azure built-in Contributor role at the subscription level, you can create a custom role with the minimum set of permissions — scoped to the minimum set of operations that Horizon Cloud requires in the associated subscription — and assign that custom role to the service principal at the subscription level. If you adopt the approach to have a separate subscription for the pod's external Unified Access Gateway configuration and select to have the gateway resources deployed into a resource group that you create and maintain, you have the option of assigning the service principal more granular, narrow-scope permissions within that separate subscription.

The overarching concept is that Horizon Cloud needs to perform certain operations in your subscription and its resource groups to successfully create and maintain the resources needed to have a pod and its gateway configurations. As a simple example, because the pod and gateway architecture require virtual machines with NICs, Horizon Cloud needs the ability to create virtual machines and NICs in your subscription and attach those NICs to subnets in the subscription's VNet. Some of the options you choose for your pod and gateway deployments determine the specific set of operations that Horizon Cloud needs to perform. You can restrict Horizon Cloud's abilities in your subscription to the minimum operations required, by following the rules described below, according to the options you are adopting for deploying a pod and for its external gateway configuration.

For details about custom roles in Microsoft Azure and the steps you take to create a custom role, see the Microsoft Azure documentation topic Custom roles for Azure resources. For details about how a role works, its structure, and the structure of the management operations, see Understand role definitions for Azure resources in the Microsoft Azure documentation. As described in that
documentation topic, a role definition is a collection of permissions. This role definition is called
the role for short. The role lists the management operations that can be performed, as well as
operations that cannot be performed, by the service principal to which that role is assigned. A
management operation is a combination of the resource and action performed on that resource.

This topic includes the following sections.

- Overview of the Available Use Cases
- When Using a Single Subscription for the Pod and Its Gateway Configurations or Using a
  Separate Subscription for the External Unified Access Gateway Configuration with
  Permissions Set at the Subscription Level
- When Using a Separate Subscription for the External Unified Access Gateway Configuration,
  Deploying into a Custom Resource Group, with Reader Role at the Subscription Level and
  Additional Required Permissions Assigned at Granular Levels

Overview of the Available Use Cases

When discussing Horizon Cloud required operations in your Microsoft Azure subscriptions and
resource groups, there are these use cases.

**Note** The role for the service principal created for the subscription that is specified for the rest
of the pod resources in the two-subscription use case must follow the same rules as needed for
the single-subscription use case.
<table>
<thead>
<tr>
<th>Use Case</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>A single subscription used by Horizon Cloud for pods and their external Unified Access Gateway configurations.</td>
<td>In this case, access must be granted to the service principal at the subscription level. The role assigned to the service principal at that level must allow the actions that Horizon Cloud needs to perform in your subscription to successfully create in that subscription the required resources and operate on those resources over time. As an example, in this case, the role must provide the ability to create the default resource groups, network security groups, virtual machines, and so on.</td>
</tr>
<tr>
<td>Two subscriptions, and you want Horizon Cloud to auto-create the gateway’s required resource groups and resources in the external gateway’s specified subscription, same as it does in the subscription for the rest of the pod resources.</td>
<td>When using this option, the service principal for each subscription must be granted access at the subscription level, with permissions that allow actions same as those for the single subscription use case described above.</td>
</tr>
</tbody>
</table>
| Two subscriptions as above, but instead of having Horizon Cloud auto-create the external gateway’s required resource groups and resources, you create a resource group in advance in that external gateway’s specified subscription, and want Horizon Cloud to deploy the external gateway’s resources into that existing resource group. | Two options for granting access to the service principal used for deploying the external gateway:  
  - Grant access at the subscription level, same as in the above case.  
  - Use the following combination:  
    - At the subscription level, grant access using the built-in Reader role.  
    - At the level of the named resource group, grant access using permissions defined in a custom role. The permissions granted at the resource-group level must provide for the operations that Horizon Cloud requires to perform in the resource group to deploy and configure the external gateway’s resources there.  
  In addition to the permissions on the resource group, Horizon Cloud needs the permissions to perform the following actions, depending on your deployment plans:  
    - If this deployment will use subnets that you create in advance on that subscription’s VNet, Horizon Cloud needs the ability to create NICs and network security groups (NSGs) on those subnets. The permissions required on the VNet that the subnet belongs to are Microsoft.Network/virtualNetworks/subnets/* and Microsoft.Network/networkSecurityGroups/*  
    - If this deployment will have Horizon Cloud generate the subnets, in addition to the above Microsoft.Network/virtualNetworks/subnets/* and Microsoft.Network/networkSecurityGroups/* |
permissions, Horizon Cloud needs the ability to create the subnets. The permission required on the VNet is `Microsoft.Network/virtualNetworks/write`.

- If your external gateway deployment will specify using a public IP address, Horizon Cloud needs the ability to create public IP addresses in the named resource group. The permission required on the named resource group is `Microsoft.Network/publicIPAddresses`.

### When Using a Single Subscription for the Pod and Its Gateway Configurations or Using a Separate Subscription for the External Unified Access Gateway Configuration with Permissions Set at the Subscription Level

For these use cases, the permissions are assigned at the subscription level. For the custom role set on the service principal that you specify in the Subscription step in the Horizon Cloud workflows, the following actions are required in the custom role definition. The * (wild card character) grants access to all operations that match the string within the listed resource provider operation. For the descriptions of the operations, see the Microsoft Azure documentation at the links listed below.

**Table 2-25. Microsoft Azure Resource Operations that Must Be Permitted in the Custom Role When Assigning Permissions at the Subscription Level**

<table>
<thead>
<tr>
<th>Operation</th>
<th>Description in the Microsoft Azure Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operation</td>
<td>Description in the Microsoft Azure Documentation</td>
</tr>
<tr>
<td>-----------------------------------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td>Operation</td>
<td>Description in the Microsoft Azure Documentation</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td>Microsoft.DBforPostgreSQL/*</td>
<td><a href="https://docs.microsoft.com/en-us/azure/role-based-access-control/resource-provider-operations#microsoftdbforpostgresql">https://docs.microsoft.com/en-us/azure/role-based-access-control/resource-provider-operations#microsoftdbforpostgresql</a></td>
</tr>
<tr>
<td>Operation</td>
<td>Description in the Microsoft Azure Documentation</td>
</tr>
<tr>
<td>------------------------------------------------</td>
<td>--------------------------------------------------</td>
</tr>
</tbody>
</table>
Table 2-25. Microsoft Azure Resource Operations that Must Be Permitted in the Custom Role When Assigning Permissions at the Subscription Level (continued)

<table>
<thead>
<tr>
<th>Operation</th>
<th>Description in the Microsoft Azure Documentation</th>
</tr>
</thead>
</table>
Table 2-25. Microsoft Azure Resource Operations that Must Be Permitted in the Custom Role When Assigning Permissions at the Subscription Level (continued)

<table>
<thead>
<tr>
<th>Operation</th>
<th>Description in the Microsoft Azure Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Microsoft.Storage/*/read</td>
<td><a href="https://docs.microsoft.com/en-us/azure/role-based-access-control/resource-provider-operations#microsoftstorage">https://docs.microsoft.com/en-us/azure/role-based-access-control/resource-provider-operations#microsoftstorage</a></td>
</tr>
</tbody>
</table>

The following JSON code block is an example to illustrate what a custom role definition named Horizon Cloud Pod might look like when it has the set of preceding operations. For a description of the properties and usage information, see the Custom role properties section in the Microsoft Azure documentation topic Custom roles for Azure resources. The ID is the unique ID of the
custom role. When you create the custom role using Azure PowerShell or Azure CLI, this ID is automatically generated when you create a new role. As described in the Tutorial: Create a custom role for Azure resources using Azure CLI, `mysubscriptionId1` is the ID of your own subscription.

```json
{
    "Name": "Horizon Cloud Pod",
    "Id": "uuid",
    "IsCustom": true,
    "Description": "Minimum set of Horizon Cloud pod required operations",
    "Actions": [
        "Microsoft.Authorization/*/read",
        "Microsoft.Compute/*/read",
        "Microsoft.Compute/availabilitySets/*",
        "Microsoft.Compute/disks/*",
        "Microsoft.Compute/images/*",
        "Microsoft.Compute/locations/*",
        "Microsoft.Compute/virtualMachines/*",
        "Microsoft.Compute/virtualMachineScaleSets/*",
        "Microsoft.Compute/snapshots/*",
        "Microsoft.DBforPostgreSQL/*",
        "Microsoft.KeyVault/*/read",
        "Microsoft.KeyVault/vaults/*",
        "Microsoft.KeyVault/vaults/secrets/*",
        "Microsoft.Network/loadBalancers/*",
        "Microsoft.Network/networkInterfaces/*",
        "Microsoft.Network/networkSecurityGroups/*",
        "Microsoft.Network/publicIPAddresses/*",
        "Microsoft.Network/virtualNetworks/read",
        "Microsoft.Network/virtualNetworks/write",
        "Microsoft.Network/virtualNetworks/checkIpAddressAvailability/read",
        "Microsoft.Network/virtualNetworks/subnets/*",
        "Microsoft.Network/virtualNetworks/virtualNetworkPeerings/read",
        "Microsoft.Resources/subscriptions/resourceGroups/*",
        "Microsoft.ResourceHealth/availabilityStatuses/read",
        "Microsoft.Resources/deployments/*",
        "Microsoft.Storage/*/read",
        "Microsoft.Storage/storageAccounts/*"
    ],
    "NotActions": [],
    "DataActions": [],
    "NotDataActions": [],
    "AssignableScopes": [
        "/subscriptions/mysubscriptionId1"
    ]
}
```
When Using a Separate Subscription for the External Unified Access Gateway Configuration, Deploying into a Custom Resource Group, with Reader Role at the Subscription Level and Additional Required Permissions Assigned at Granular Levels

For this use case, you can assign the built-in Reader role to the service principal at the subscription level, and then grant access at the level of the named resource group using a custom role that specifies the permissions in the following table. Some additional permissions on subnets and on the VNet are required, depending on your planned deployment options:

- If this external gateway deployment will use subnets that you create in advance, Horizon Cloud needs the ability to create NICs and network security groups (NSGs) on those subnets. The permissions required on the VNet that the subnet belongs to are Microsoft.Network/virtualNetworks/subnets/* and Microsoft.Network/networkSecurityGroups/*

- If this external gateway deployment will have Horizon Cloud generate the subnets, in addition to the above Microsoft.Network/virtualNetworks/subnets/* and Microsoft.Network/networkSecurityGroups/* permissions, Horizon Cloud needs the ability to create the subnets. The permission required on the subscription's VNet is Microsoft.Network/virtualNetworks/write

- If your deployment will specify using a public IP address for the external gateway configuration, Horizon Cloud needs the ability to create public IP addresses in the named resource group. The permission required on the named resource group is Microsoft.Network/publicIPAddresses

The following permitted operations are required in the named resource group. The * (wild card character) grants access to all operations that match the string within the listed resource provider operation. For the descriptions of the operations, see the Microsoft Azure documentation at the links listed below.

Table 2-27. Microsoft Azure Resource Operations that Must Be Permitted on the Specified Resource Group

<table>
<thead>
<tr>
<th>Operation</th>
<th>Description in the Microsoft Azure Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Operation</td>
<td>Description in the Microsoft Azure Documentation</td>
</tr>
<tr>
<td>-------------------------------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td>Operation</td>
<td>Description in the Microsoft Azure Documentation</td>
</tr>
<tr>
<td>-----------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td>Microsoft.DBforPostgreSQL/*</td>
<td><a href="https://docs.microsoft.com/en-us/azure/role-based-access-control/resource-provider-operations#microsoftdbforpostgresql">https://docs.microsoft.com/en-us/azure/role-based-access-control/resource-provider-operations#microsoftdbforpostgresql</a></td>
</tr>
<tr>
<td>Operation</td>
<td>Description in the Microsoft Azure Documentation</td>
</tr>
<tr>
<td>------------------------------------------------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td>Microsoft.Network/publicIPAddresses/*, if your deployment will specify using a public IP address for the external gateway deployment.</td>
<td><a href="https://docs.microsoft.com/en-us/azure/role-based-access-control/resource-provider-operations#microsoftnetwork">https://docs.microsoft.com/en-us/azure/role-based-access-control/resource-provider-operations#microsoftnetwork</a></td>
</tr>
<tr>
<td>Operation</td>
<td>Description in the Microsoft Azure Documentation</td>
</tr>
<tr>
<td>--------------------------------------------------------------------------</td>
<td>--------------------------------------------------</td>
</tr>
</tbody>
</table>
Table 2-27. Microsoft Azure Resource Operations that Must Be Permitted on the Specified Resource Group (continued)

<table>
<thead>
<tr>
<th>Operation</th>
<th>Description in the Microsoft Azure Documentation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Microsoft.Storage/*/read</td>
<td><a href="https://docs.microsoft.com/en-us/azure/role-based-access-control/resource-provider-operations#microsoftstorage">https://docs.microsoft.com/en-us/azure/role-based-access-control/resource-provider-operations#microsoftstorage</a></td>
</tr>
</tbody>
</table>

Subscription-Related Information for the Horizon Cloud Pod Deployment Wizard

The Horizon Cloud pod deployment wizard requires you to provide the following pieces of information from your Microsoft Azure subscription. If you are deploying the external gateway into its own subscription, separate from the pod's, the deployer requires this information for both subscriptions.

**Important** You must obtain the application key at the moment you generate it in the Microsoft Azure portal. For information, see Create the Required Service Principal Needed by the Horizon Cloud Pod Deployer by Creating an Application Registration. You can obtain the other pieces of information at any time by logging in to your Microsoft Azure portal using your Microsoft Azure account credentials.

The IDs are UUIDs, in the form 8-4-4-4-12. These IDs and key described in the following table are used in the first step of the pod deployment wizard.
### Required Value | How to Collect | Your Values
--- | --- | ---
**Environment** | You determine the Microsoft Azure cloud environment when you register for your Microsoft Azure subscription. At that point in time, your account and subscription is created within the specific Microsoft Azure environment. |  
**Subscription ID** | In the Microsoft Azure portal, navigate to your subscription's settings screen by clicking *All services* in the Microsoft Azure portal's main navigation bar, clicking *Subscriptions*, and then clicking the name of the subscription that you will use with the pod. Locate the displayed subscription ID. |  
**Directory ID** | In the Microsoft Azure portal, click > Properties (under *Manage*). |  
**Application ID** | In the Microsoft Azure portal, click > App registrations, and then click the application registration that you created for Horizon Cloud using the steps in Create the Required Service Principal Needed by the Horizon Cloud Pod Deployer by Creating an Application Registration. |  
**Application Key** | Obtain the key when you generate it in the Microsoft Azure portal. See Create the Required Service Principal Needed by the Horizon Cloud Pod Deployer by Creating an Application Registration. |  

**Convert a Certificate File to the PEM Format Required for Pod Deployment**

The Unified Access Gateway capability in your pod requires SSL for client connections. When you want the pod to have a Unified Access Gateway configuration, the pod deployment wizard requires a PEM-format file to provide the SSL server certificate chain to the pod's Unified Access Gateway configuration. The single PEM file must contain the full entire certificate chain including the private key: the SSL server certificate, any necessary intermediate CA certificates, the root CA certificate, and private key.

For additional details about certificate types used in Unified Access Gateway, see the topic titled Selecting the Correct Certificate Type in the [Unified Access Gateway product documentation](#).
In the pod deployment wizard step for the gateway settings, you upload a certificate file. During the deployment process, this file is submitted in to the configuration of the deployed Unified Access Gateway instances. When you perform the upload step in the wizard interface, the wizard verifies that the file you upload meets these requirements:

- The file can be parsed as PEM-format.
- It contains a valid certificate chain and a private key.
- That private key matches the public key of the server certificate.

If you do not have a PEM-format file for your certificate information, you must convert your certificate information into a file that meets those above requirements. You must convert your non-PEM-format file into PEM format and create a single PEM file that contains the full certificate chain plus private key. You also need to edit the file to remove extra information, if any appears, so that the wizard will not have any issues parsing the file. The high-level steps are:

1. Convert your certificate information into PEM format and create a single PEM file that contains the certificate chain and the private key.
2. Edit the file to remove extra certificate information, if any, that is outside of the certificate information between each set of "-----BEGIN CERTIFICATE-----" and "-----END CERTIFICATE-----" markers.

The code examples in the following steps assume you are starting with a file named `mycaservercert.pfx` that contains the root CA certificate, intermediate CA certificate information, and private key.

**Prerequisites**

- Verify that you have your certificate file. The file can be in PKCS#12 (.p12 or .pfx) format or in Java JKS or JCEKS format.

**Important** All certificates in the certificate chain must have valid time frames. The Unified Access Gateway VMs require that all of the certificates in the chain, including any intermediate certificates, have valid time frames. If any certificate in the chain is expired, unexpected failures can occur later as the certificate is uploaded to the Unified Access Gateway configuration.

- Familiarize yourself with the `openssl` command-line tool that you can use to convert the certificate. See [https://www.openssl.org/docs/apps/openssl.html](https://www.openssl.org/docs/apps/openssl.html).

- If the certificate is in Java JKS or JCEKS format, familiarize yourself with the Java `keytool` command-line tool to first convert the certificate to .p12 or .pks format before converting to .pem files.
Procedure

1. If your certificate is in Java JKS or JCEKS format, use `keytool` to convert the certificate to `.p12` or `.pks` format.

   **Important** Use the same source and destination password during this conversion.

2. If your certificate is in PKCS#12 (.p12 or .pfx) format, or after the certificate is converted to PKCS#12 format, use `openssl` to convert the certificate to a `.pem` file.

   For example, if the name of the certificate is `mycaservercert.pfx`, you can use the following commands to convert the certificate:

   ```
   openssl pkcs12 -in mycaservercert.pfx -nokeys -out mycaservercertchain.pem
   openssl pkcs12 -in mycaservercert.pfx -nodes -nocerts -out mycaservercertkey.pem
   ```

   The first line above obtains the certificates in `mycaservercert.pfx` and writes them in PEM format to `mycaservercertchain.pem`. The second line above obtains the private key from `mycaservercert.pfx` and writes it in PEM format to `mycaservercertkey.pem`.

3. (Optional) If the private key is not in RSA format, convert the private key to the RSA private key format.

   The Unified Access Gateway instances require the RSA private key format. To check if you need to run this step, look at your PEM file and see if the private key information starts with

   ```
   -----BEGIN PRIVATE KEY-----
   ```

   If the private key starts with that line, then you should convert the private key to the RSA format. If the private key starts with `-----BEGIN RSA PRIVATE KEY-----`, you do not have to run this step to convert the private key.

   To convert the private key to RSA format, run this command.

   ```
   openssl rsa -in mycaservercertkey.pem -check -out mycaservercertkeyrsa.pem
   ```

   The private key in the PEM file is now in RSA format (`-----BEGIN RSA PRIVATE KEY-----` and `-----END RSA PRIVATE KEY-----`).

4. Combine the information in the certificate chain PEM file and private key PEM file to make a single PEM file.

   The example below shows a sample where the contents of `mycaservercertkeyrsa.pem` is first (the private key in RSA format), followed by the contents from `mycaservercertchain.pem`, which is your primary SSL certificate, followed by one intermediate certificate, followed by the root certificate.

   ```
   -----BEGIN CERTIFICATE-----
   .... (your primary SSL certificate)
   -----END CERTIFICATE-----
   ```
Note The server certificate should come first, followed by any intermediate ones, and then the trusted root certificate.

If there are any unnecessary certificate entries or extraneous information between the BEGIN and END markers, edit the file to remove those.

Results

The resulting PEM file meets the requirements of the pod deployment wizard.

Deploy a Horizon Cloud Pod into Microsoft Azure

You run the pod deployment wizard to deploy the components that together make up a pod and its gateway configurations. The pod’s connector component pairs with Horizon Cloud so that you can use your Microsoft Azure capacity with Horizon Cloud.

The deployer uses the information you provide in each step of the wizard to determine how to configure the pod. After you provide the requested information in a particular step, proceed to the next step by clicking Next.

Caution The IP addresses mentioned in these steps are examples. You should use the address ranges that meet your organization’s needs. For each step that mentions an IP address range, substitute ones that are applicable for your organization.

Prerequisites

Before you start the pod deployment wizard, verify that you have the required items. The items you need to provide in the wizard vary according to the pod configuration options you want. For the prerequisites, see Prerequisites for Running the Pod Deployment Wizard.

Pod configuration options include:

- Selecting existing subnets that you create in advance or having the pod deployer automatically create the subnets
- Selecting whether to enable high availability for the pod. If you deploy the pod without high availability enabled, you can later edit the pod to enable it.
- Having the deployment process create a VMware Workspace ONE® Access™ tenant.
Deploying with an external or internal Unified Access Gateway configuration, or deploying with both. If you deploy with only one type of gateway configuration, you can later edit the pod to add the other, non-configured type.

<table>
<thead>
<tr>
<th>If you deploy with the Unified Access Gateway configuration as...</th>
<th>You can later edit the pod to add...</th>
</tr>
</thead>
<tbody>
<tr>
<td>External</td>
<td>Internal</td>
</tr>
<tr>
<td>Internal</td>
<td>External</td>
</tr>
</tbody>
</table>

Deploying with an external Unified Access Gateway configuration in its own VNet, separate from the pod's VNet.

Deploying with an external Unified Access Gateway configuration in its own subscription, separate from the pod's subscription. Because VNets do not span subscriptions, this option is a special scenario of the separate VNet case — when the external gateway is deployed using its own subscription, it also means it is in its own VNet.

Deploying with the option for RADIUS two-factor authentication configured on the pod’s gateway configurations. If you deploy without RADIUS settings configured on the pod’s gateway configurations, you can later edit the pod to add the other, non-configured type.

For an external Unified Access Gateway configuration, you can optionally select to not have a public IP address on the configuration’s load balancer. If you select the wizard option to not have a public IP address on the load balancer, you must specify in the wizard an IP value that you have mapped to an FQDN in your DNS server. That FQDN is the one that will be used in your end users' Horizon clients for PCoIP connections to this gateway. In the deployment process, the deployer will configure that IP address in the Unified Access Gateway's Horizon configuration settings. In the Unified Access Gateway documentation, this IP value is referred to as the PCoIP External URL. Even though the Unified Access Gateway documentation refers to it as a URL, the entered value must be an IP address. You map this IP address to an FQDN in your DNS, which is the FQDN used in your end users' Horizon clients to establish their PCoIP sessions with the pod's external Unified Access Gateway configuration.

Caution You cannot later edit the deployed pod to change this IP address setting for the external gateway's load balancer. Therefore, ensure that you input the public IP address in the deployment wizard that matches your DNS mapping with the FQDN, and the FQDN matches the one in the certificate that you upload in the deployment wizard.

Procedure

1 Prerequisites for Running the Pod Deployment Wizard

Before you run the pod deployment wizard, verify that your environment satisfies these prerequisites. You must have the following items so that you can provide the requested values in the pod deployment wizard and proceed through the wizard.
2 Start the Pod Deployment Wizard
   When deploying your very first pod into Microsoft Azure, you start the pod deployment wizard using the Add Cloud Capacity feature on the Horizon Cloud Administration Console’s Getting Started page.

3 Specify the Microsoft Azure Subscription Information for the New Horizon Cloud Pod
   In this step of the pod deployment wizard, you provide the Microsoft Azure subscription information that you want to use for this pod.

4 Specify Pod Configuration Information for the Horizon Cloud Pod You Are Deploying Into Microsoft Azure
   In the Pod Setup step of the pod deployment wizard, you specify details such as the name of the pod, as well as networking information.

5 Specify the Horizon Cloud Pod’s Gateway Configuration
   In this step of the wizard, specify the information required to deploy the pod with a gateway configured. Unified Access Gateway provides the gateway environment for a pod deployed into Microsoft Azure. When deploying the new pod, you can choose to have either an external or internal gateway configuration — or have both types on the same pod. By default, when this wizard step displays, the external gateway configuration is selected.

6 Validate and Proceed, and then Start the Pod Deployment Process
   After you click Validate & Proceed, the system verifies your specified values. If everything validates, the wizard displays a summary of the information for your review. Then you start the deployment process.
Prerequisites for Running the Pod Deployment Wizard

Before you run the pod deployment wizard, verify that your environment satisfies these prerequisites. You must have the following items so that you can provide the requested values in the pod deployment wizard and proceed through the wizard.

**Important** Before launching the pod deployment wizard and starting to deploy your pod, in addition to the requirements below, you must be aware of the following key points:

- Starting with the July 2020 service release, in brand new environments, new pods are required to deploy with at least one gateway configuration. If your customer account was created prior to the July 2020 release, but you have not yet deployed your first pod, deployment of that first pod will require configuring at least one gateway configuration at the time of pod deployment.

- Successful pod deployment requires that none of the Microsoft Azure Policies that you or your IT team have set in your Microsoft Azure environment block, deny, or restrict creation of the pod's components. Also you must verify that your Microsoft Azure Policies Built-in Policy definitions do not block, deny, or restrict creation of the pod's components. As an example, you and your IT team must verify that none of your Microsoft Azure Policies block, deny, or restrict creation of components on Azure storage account. For information about Azure Policies, see the [Azure Policy documentation](#).

- The pod deployer requires that your Azure storage account allow for the deployer to use the Azure StorageV1 and StorageV2 account types. Ensure that your Microsoft Azure Policies do not restrict or deny the creation of content requiring the Azure StorageV1 and StorageV2 account types.

- As part of the pod and gateway deployment processes, Horizon Cloud creates resource groups (RGs) in your Microsoft Azure subscription that do not have tags on them, including the initial resource group that is created for the temporary jump box that orchestrates those deployment processes. Pod deployment will fail if you try to deploy a pod into a Microsoft Azure subscription that has any type of resource tag requirement at the time of deployment, or at the time of pod updates or adding a gateway configuration to a pod. You must verify that your Microsoft Azure Policies allows creation of the pod's untagged resource groups in the target subscription. For the list of RGs that the deployer creates, see the Administration Guide's [Resource Groups Created For a Pod Deployed In Microsoft Azure](#) topic.

- All cloud-connected pods must have line-of-sight to the same set of Active Directory domains at the time you deploy those pods.

Prerequisites for All Deployments

- Verify that all of the preparatory tasks are completed, as described in [Preparing to Deploy a Horizon Cloud Pod Into Microsoft Azure](#).

- Verify that you have the subscription information, as described in [Subscription-Related Information for the Horizon Cloud Pod Deployment Wizard](#).
- Verify that you have an existing virtual network in your Microsoft Azure subscription, and in the region in which you are deploying the pod, as described in Configure the Required Virtual Network in Microsoft Azure.

  **Important** Not all Microsoft Azure regions support GPU-enabled virtual machines. If you want to use the pod for GPU-capable desktops or remote applications, ensure that the Microsoft Azure region you select for the pod provides for those NV-series VM types that you want to use and which are supported in this Horizon Cloud release. See the Microsoft documentation at https://azure.microsoft.com/en-us/regions/services/ for details.

- Verify that your VNet is configured to point to a DNS that can resolve external addresses. The pod deployer must be able to reach external addresses in the Horizon Cloud control plane to securely download the pod software into your Microsoft Azure environment.

- Verify that the pod deployer's DNS, ports, and protocols requirements are met, as described in DNS Requirements for a Horizon Cloud Pod in Microsoft Azure and Ports and Protocols Requirements for a Horizon Cloud Pod at the September 2019 Release's Manifest or Later.

- If you require use of a proxy for outbound Internet access, verify you have the networking information for your proxy configuration and the authentication credentials it requires, if any. The pod deployment process requires outbound Internet access.

- Verify that you have the information for at least one NTP server that you want the pod to use for time synchronization. The NTP server can be a public NTP server or your own NTP server that you set up for this purpose. The NTP server you specify must be reachable from the virtual network you configured. When you plan to use an NTP server using its domain name instead of a numeric IP address, also ensure that the DNS configured for the virtual network can resolve the NTP server's name.

- If you do not want the deployer to automatically create the subnets it needs, verify that the required subnets have been created in advance and exist on the VNet. For the steps to create the required subnets in advance, see In Advance of Pod Deployment, Create the Horizon Cloud Pod’s Required Subnets on your VNet in Microsoft Azure and When Using Existing Subnets for a Horizon Cloud Pod in Microsoft Azure.

  **Caution** The subnets you manually create on your VNet in advance for the pod deployment must remain empty. Do not put any resources on these subnets or otherwise use any of the IP addresses. If an IP address is already in use on the subnets, the pod might fail to deploy.

- If you are going to have the deployer create the required subnets, verify that you know the address ranges you are going to enter into the wizard for the management subnet, desktop subnet, and DMZ subnet. The DMZ subnet is required when you want the external Unified Access Gateway configuration. Also verify that those ranges do not overlap. You enter the address ranges using CIDR notation (classless inter-domain routing notation). The wizard will display an error if the entered subnet ranges overlap. For the management subnet range, a
CIDR of at least /27 is required. For the DMZ subnet range, a CIDR of at least /28 is required. If you want to keep the management and DMZ subnet ranges co-located, you can specify the DMZ subnet range similar to the management subnet with an IP specified. For example, if the management subnet is 192.168.8.0/27, a matching DMZ subnet would be 192.168.8.32/27.

**Important** The CIDRs you enter in the wizard’s fields must be defined so that each combination of prefix and bit mask results in an IP address range having the prefix as the starting IP address. Microsoft Azure requires that the CIDR prefix be the start of the range. For example, a correct CIDR of 192.168.182.48/28 would result in an IP range of 192.168.182.48 to 192.168.182.63, and the prefix is the same as the starting IP address (192.168.182.48). However, an incorrect CIDR of 192.168.182.60/28 would result in an IP range of 192.168.182.48 to 192.168.182.63, where the starting IP address is not the same as the prefix of 192.168.182.60. Ensure that your CIDRs result in IP address ranges where the starting IP address matches the CIDR prefix.

- If you are going to have the deployer create the required subnets, verify that subnets with those address ranges do not already exist on the VNet. In this scenario, the deployer itself will automatically create the subnets using the address ranges you provide in the wizard. If the wizard detects subnets with those ranges already exist, the wizard will display an error about overlapping addresses and will not proceed further. If your VNet is peered, also verify that the CIDR address spaces that you plan to enter in the wizard are already contained in the VNet’s address space.

**Prerequisites for the Unified Access Gateway Configurations**

If you are planning to have the pod use a Unified Access Gateway configuration, you must provide:

- The fully qualified domain name (FQDN) which your end users will use to access the service. Starting with the new pod manifest made available in the July 2020 quarterly service release, when deploying both of the gateway configurations on a pod, you are required to specify the same FQDN for both gateway configurations. After the pod is deployed with both the external and internal gateway configuration, you must configure the incoming end-user client traffic to route to the appropriate load balancer. The goal is to set up the routing so that client traffic from the Internet is routed to the external gateway’s Microsoft Azure Public Load Balancer and client traffic from your intranet is routed to the internal gateway’s Microsoft Azure Internal Load Balancer. Because both gateways will have the same FQDN, you configure Split DNS (Split Domain Name System) to resolve the gateway address either to the external gateway or internal gateway depending on the origin network of the end-user client’s DNS query. Then the same FQDN used in the end-user client can route to the external gateway when the client is on the Internet and route to the internal gateway when the client is on your internal network.

**Important** This FQDN cannot contain underscores. In this release, connections to the Unified Access Gateway instances will fail when the FQDN contains underscores.
A signed SSL server certificate (in PEM format) based on that FQDN. The Unified Access Gateway capabilities require SSL for client connections, as described in the Unified Access Gateway product documentation. The certificate must be signed by a trusted Certificate Authority (CA). The single PEM file must contain the full entire certificate chain with the private key. For example, the single PEM file must contain the SSL server certificate, any necessary intermediate CA certificates, the root CA certificate, and private key. OpenSSL is a tool you can use to create the PEM file.

**Important**  All certificates in the certificate chain must have valid time frames. The Unified Access Gateway VMs require that all of the certificates in the chain, including any intermediate certificates, have valid time frames. If any certificate in the chain is expired, unexpected failures can occur later as the certificate is uploaded to the Unified Access Gateway configuration.

If you are deploying with an external Unified Access Gateway configuration, you must specify a DMZ (demilitarized zone) subnet. You can provide for this DMZ subnet by one of two ways:

- Creating the DMZ subnet in advance on the VNet. With this method, you also have to create the management and desktop tenant subnets in advance. See the steps in In Advance of Pod Deployment, Create the Horizon Cloud Pod's Required Subnets on your VNet in Microsoft Azure.

- Having the deployer automatically create the DMZ subnet during deployment. With this method, you must have the address range you are going to enter into the wizard for the DMZ subnet and verify that the range does not overlap with the ranges for the management and desktop tenant subnets. You enter the address ranges using CIDR notation (classless inter-domain routing notation). The wizard will display an error if the entered subnet ranges overlap. For the DMZ subnet range, a CIDR of at least /28 is required. If you want to keep the management and DMZ subnet ranges co-located, you can specify the DMZ subnet range the same as the management subnet with an IP specified. For example, if the management subnet is 192.168.8.0/27, a matching DMZ subnet would be 192.168.8.32/27. Also see the important note in Prerequisites for All Deployments about ensuring the IP address range has a combination of prefix and bit mask that results in the range having the prefix as the starting IP address.

If you are deploying with an external Unified Access Gateway configuration and you want to disable having a public IP address for the configuration's load balancer, you must specify an IP address that you have mapped in your DNS settings to the FQDN which your end users will use for PCoIP connections in their Horizon clients.

For more information about the PEM file considerations required by Unified Access Gateway, see Convert a Certificate File to the PEM Format Required for Pod Deployment.
Prerequisites When Deploying With an External Unified Access Gateway Configuration Using its Own VNet or Subscription Separate from the Pod's VNet or Subscription

Along with the above prerequisites when deploying with a Unified Access Gateway configuration, these prerequisites are specific to the use case of deploying the external gateway in its own VNet or own subscription. Using its own subscription is a special case of using its own VNet, because the separate subscription must have its own VNet, because VNets are scoped to a subscription.

- The VNet for the gateway must be peered with the pod's VNet.
- Verify that either the required subnets have been created in advance and exist on the VNet, or that the CIDR address spaces that you plan to enter in the wizard are already contained in the VNet's address space. Because the VNets are peered, the deployer will not be able to expand the VNet automatically if you enter into the wizard CIDR address spaces that are not already contained in the VNet's address space. If that happens, the deployment process will fail.

**Tip** The best practice is to create the subnets in advance. For the steps to create the required subnets in advance, see In Advance of Pod Deployment, Create the Horizon Cloud Pod's Required Subnets on your VNet in Microsoft Azure and When Using Existing Subnets for a Horizon Cloud Pod in Microsoft Azure.

- If you are using a separate subscription for the external gateway, verify that you have the subscription information, as described in Subscription-Related Information for the Horizon Cloud Pod Deployment Wizard.
- If you are using a separate subscription for the external gateway and are planning to deploy the gateway into a named resource group that you create instead of having the deployer auto-create the resource group, verify that you have created that resource group in that subscription. You will select that resource group by name in the wizard. Also verify that you have granted the required access to that resource group for the deployer to operate in it, as described in Operations Required by Horizon Cloud in Your Microsoft Azure Subscriptions.

Prerequisites When Deploying With a Two-Factor Authentication Configuration

If you are planning to use the two-factor authentication capability, or use it with an on-premises two-factor authentication server, verify that you have the following information used in your authentication server's configuration, so that you can provide it in the appropriate fields in the pod deployment wizard. If you have both a primary and secondary server, obtain the information for each of them.

- IP address or DNS name of the authentication server
- The shared secret that is used for encryption and decryption in the authentication server's protocol messages
- Authentication port numbers, typically the 1812 UDP port.
Authentication protocol type. The authentication types include PAP (Password Authentication Protocol), CHAP (Challenge Handshake Authentication Protocol), MSCHAP1, MSCHAP2 (Microsoft Challenge Handshake Authentication Protocol, version 1 and 2).

**Note** Check your RADIUS vendor’s documentation for the authentication protocol that your RADIUS vendor recommends and follow their indicated protocol type. The pod’s capability to support two-factor authentication with RADIUS is provided by the Unified Access Gateway instances, and Unified Access Gateway supports PAP, CHAP, MSCHAP1, and MSCHAP2. PAP is generally less secure than MSCHAP2. PAP is also a simpler protocol than MSCHAP2. As a result, even though most RADIUS vendors are compatible with the simpler PAP protocol, some RADIUS vendors are not as compatible with the more secure MSCHAP2.

Start the Pod Deployment Wizard

When deploying your very first pod into Microsoft Azure, you start the pod deployment wizard using the Add Cloud Capacity feature on the Horizon Cloud Administration Console’s Getting Started page.

**Note** Login authentication into the cloud-based console relies on authenticating account credentials either with the My VMware account system or VMware Cloud Services system. If those systems are experiencing a system outage and cannot take authentication requests, you will not be able to log in to the console during that time period. If you encounter issues logging in to the console's first login screen, check the Horizon Cloud System Status page at https://status.horizon.vmware.com to see the latest system status. On that page, you can also subscribe to receive updates.

Prerequisites

Verify that you have met the prerequisites described in Prerequisites for Running the Pod Deployment Wizard.

Procedure

1. Log in to the administrative console at https://cloud.horizon.vmware.com using your preferred method.
   - In the **My VMware Credentials** section of the login page, enter My VMware account's credentials. The account credentials are the primary email address, such as user@example.com, and the password that are set in the account's profile. This choice sends the authentication request to the Horizon Cloud control plane.
   - In the **VMware Cloud Services** section of the login page, click **VMWARE CLOUD LOGIN**. Clicking that button redirects the authentication request to VMware Cloud Services, to authenticate you according to your organization's configuration there. Use this method if your organization has asked you to access their Horizon Cloud tenant using VMware Cloud Services.

   The following screenshot illustrates logging in by entering My VMware account's credentials.
If you have not previously accepted the Horizon Cloud terms of service using those My VMware credentials, a terms of service notification box appears after you click the Login button. Accept the terms of service to continue.

When your login is successfully authenticated, the console displays in your browser. When you have no existing pods, the Getting Started wizard is displayed by default with the Capacity section expanded and the Add Cloud Capacity row.
2  In the Add Cloud Capacity row, click **Manage > Add Pod.**

A selection window appears where you can select the cloud into which to deploy this pod.

3  Click **Select** for the Microsoft Azure cloud.

   The Add Cloud Capacity wizard opens to its first step.
4 Specify the subscription to use for this pod by following the steps in Specify the Microsoft Azure Subscription Information for the New Horizon Cloud Pod.

Specify the Microsoft Azure Subscription Information for the New Horizon Cloud Pod

In this step of the pod deployment wizard, you provide the Microsoft Azure subscription information that you want to use for this pod.

Prerequisites

- Verify that you have met the prerequisites described in Deploy a Horizon Cloud Pod into Microsoft Azure.
- For this wizard step, verify that you have the subscription-related information as described in Subscription-Related Information for the Horizon Cloud Pod Deployment Wizard.
- Complete the steps in Start the Pod Deployment Wizard.
Procedure

1. On the wizard's first step, specify the subscription to use for this pod by selecting the name of a previously entered subscription or entering new subscription information.

Add Microsoft Azure Capacity

Choose the Microsoft Azure subscription you want to apply or add a new one.

Pod Subscription

Apply Subscription: Add New
Subscription Name: 
Environment: Select
Subscription ID: 
Directory ID: 
Application ID: 
Application Key: 

Use a Different Subscription for External Gateway: 

CANCEL ADD
If you select an existing subscription, the step is populated with that subscription's information that was previously entered into the system.

**Note** You might wonder why there would be previously entered subscription information when you are deploying a pod from the initial Getting Started page. Previously entered subscription information is possible in situations such as the following examples:

- You start the wizard, enter subscription information in this first wizard step, and click Add to submit the subscription information to the system and progress forward in the wizard. Then on a subsequent step, you cancel out of the wizard before completing all of the steps. In this situation, the system has saved the subscription information you entered in this first wizard step after you clicked Add. Even though you cancel out of the wizard in a subsequent step, the system retains that previously entered subscription information.

- You used this Horizon Cloud customer account record before, deploying first and subsequent pods for that account record, and then at some point in time, deleted those pods. When you log back in with the credentials that are associated with your Horizon Cloud customer account record, subscription information that was previously entered is still associated with that customer record, and previous subscription names are displayed in the drop-down list.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Apply Subscription</td>
<td>Select the name of a previously entered subscription or select Add New to enter new subscription information.</td>
</tr>
<tr>
<td>Subscription Name</td>
<td>When providing new subscription information, enter a friendly name so you can identify this subscription from other previously entered subscriptions. The name must start with a letter and contain only letters, dashes, and numbers.</td>
</tr>
<tr>
<td>Environment</td>
<td>Select the cloud environment associated with your subscription, for example: Azure - Commercial, for the standard global Microsoft Azure cloud regions Azure - China, for the Microsoft Azure in China cloud Azure - US Government, for the Microsoft Azure US Government cloud</td>
</tr>
<tr>
<td>Subscription ID</td>
<td>Enter your cloud capacity subscription ID (in UUID form). This subscription ID must be valid for the environment you selected. For Microsoft Azure, you can obtain this UUID from your Microsoft Azure portal's Subscriptions area.</td>
</tr>
<tr>
<td>Directory ID</td>
<td>Enter your Microsoft Azure AD Directory ID (in UUID form). For Microsoft Azure, you can obtain this UUID from your Microsoft Azure Active Directory properties in the Microsoft Azure portal.</td>
</tr>
<tr>
<td>Application ID</td>
<td>Enter the application ID (in UUID form) associated with the service principal you created in the Microsoft Azure portal. Creating an application registration and its associated service principal in your Microsoft Azure Active Directory is a prerequisite.</td>
</tr>
<tr>
<td>Option</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------------------------------</td>
<td>------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>Application Key</strong></td>
<td>Enter the key value for the service principal's authentication key that you created in the Microsoft Azure portal. Creating this key is a prerequisite.</td>
</tr>
<tr>
<td><strong>Use a Different Subscription for External Gateway</strong></td>
<td>Enable this toggle when you want to deploy an external Unified Access Gateway configuration into its own subscription, separate from the pod's subscription. Using separate subscriptions for the external gateway gives your organization the flexibility to assign separate teams control over those subscriptions, depending on their area of expertise. It allows for more granular access control for which people in your organization can access the pod's assets in its subscription's resource groups and which people can access the gateway's assets. When this toggle is turned on, the fields for entering the gateway's subscription information are displayed. Specify the information in those fields as you did for the pod's subscription.</td>
</tr>
</tbody>
</table>

**Important** In this screen, you cannot delete previously entered subscription values associated with a particular **Subscription Name**. Even though this occurrence is rare, you might imagine a situation where you:

a. Set up the subscription-related pieces in Microsoft Azure.
b. Start the Add Cloud Capacity wizard, enter those subscription values in the first step, and progress on to the next wizard step.
c. However, upon reading the networking values requested in the next wizard step, you cancel out of this wizard and open a new browser tab to go into the Microsoft Azure portal and adjust your networking configuration to meet the prerequisites.
d. While in the Microsoft Azure portal, you then also decide to make a new application registration to have a service provider with a different name.
e. You return to the browser that has the Getting Started page and restart the Add Cloud Capacity wizard.

At this point, your previously entered subscription name is still in the **Apply Subscription** drop-down list. However, if you select that name, all of the fields are prepopulated with the previous values, including the old application ID, and there is not a way to change the values in the screen or edit or delete that subscription name to start over with it. If this happens to you, cancel out of the wizard, restart the wizard, and on this first step, create a brand new subscription name by selecting **Add New**, enter the current values you want to use, and proceed forward.

The following screenshot is an example with the main subscription details completed.
Choose the Microsoft Azure subscription you want to apply or add a new one.

Pod Subscription

<table>
<thead>
<tr>
<th><strong>Apply Subscription:</strong>*</th>
<th>Add New</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Subscription Name:</strong>*</td>
<td>mysub1</td>
</tr>
<tr>
<td><strong>Environment:</strong>*</td>
<td>Azure - Commercial</td>
</tr>
<tr>
<td><strong>Subscription ID:</strong>*</td>
<td></td>
</tr>
<tr>
<td><strong>Directory ID:</strong>*</td>
<td></td>
</tr>
<tr>
<td><strong>Application ID:</strong>*</td>
<td></td>
</tr>
<tr>
<td><strong>Application Key:</strong>*</td>
<td></td>
</tr>
</tbody>
</table>

2 Proceed to the next wizard step.

When you click the button to proceed to the next step, the system verifies the validity of all of the specified values and whether they are appropriately related to each other, such as:

- Is the specified subscription ID valid in the selected environment.
- Are the specified directory ID, application ID, and application key valid in that subscription.
- Is the application's service principal for the specified application ID assigned a role that permits all of the operations that the deployment process requires for the type of deployment you are doing. For a description of the service principal and its role requirements, see the topic [Create the Required Service Principal by Creating an Application Registration](#) and [Operations Required by Horizon Cloud in Your Microsoft Azure Subscriptions](#).

If you see an error message about correcting values, at least one of the values is invalid either by not existing in your subscription or not having a valid relationship with another of the values. Here is a list of some, though not necessarily all, situations that can result in that error message:

- If you specified a **Directory ID** that is in your subscription but you specified an **Application ID** value that is in a different directory.
If the specified service principal’s assigned role does not permit the operations that the pod deployer requires.

**Important** More than one piece might be invalid when that error message appears. If you see that error message, verify the subscription-related information that you collected and the configuration of the service principal.

3 Specify pod details and networking information by following the steps in Specify Pod Configuration Information for the Horizon Cloud Pod You Are Deploying Into Microsoft Azure.

**Specify Pod Configuration Information for the Horizon Cloud Pod You Are Deploying Into Microsoft Azure**

In the Pod Setup step of the pod deployment wizard, you specify details such as the name of the pod, as well as networking information.

**Caution** The IP addresses mentioned in these steps are examples. You should use the address ranges that meet your organization's needs. For each step that mentions an IP address range, substitute ones that are applicable for your organization.

**Prerequisites**

Verify that you have met the prerequisites described in Prerequisites for Running the Pod Deployment Wizard.

If you are going to have the deployment process automatically create the required subnets, verify that the CIDR address ranges that you plan to specify in the wizard fields for those subnets are not already being used by existing subnets on your VNet in Microsoft Azure.

If you have created subnets in advance for use with this pod, verify those subnets have no resources attached to them and verify that the subnet you created to use for the management subnet has the Microsoft.SQL service configured as a service endpoint for that subnet. The pod deployment wizard will validate that the Microsoft.SQL service is configured as a service endpoint on the management subnet.

**Caution** These subnets you create on your VNet for a pod deployment must be empty. You can create the subnets prior to deploying the pod, but do not put any resources on those subnets or otherwise use any of the IP addresses. If an IP address is already in use on the subnets, the pod might fail to deploy.

**Procedure**

1 In this step of the wizard, provide details about the pod and the required networking information.

   The following screenshot is an example of the step when it is initially displayed.
## Add Microsoft Azure Capacity

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Pod Name</strong></td>
<td>Enter a friendly name for this pod. This name is used in the administrative console to identify this pod from your other pods.</td>
</tr>
<tr>
<td><strong>Location</strong></td>
<td>Select an existing city name or click Add to specify a new city. The system groups your pods according to city name, and depicts them on the console's Dashboard page's Horizon Global Footprint map. When you click Add, start typing the name of a city. The system automatically displays world city names in its backend geography lookup table that match your entered characters, and you can choose a city from that list. <strong>Note</strong> You must select a city from the system's autocomplete list.</td>
</tr>
<tr>
<td>Option</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Microsoft Azure Region</td>
<td>Select the physical geographic Microsoft Azure region into which you want the pod to be deployed. The available regions are determined by the previously selected Microsoft Azure environment. Consider choosing the region based on its proximity to the end users you intend to serve with this pod. Nearer proximity would provide lower latency. Important Not all Microsoft Azure regions support GPU-enabled virtual machines. If you want to use the pod for GPU-capable desktops or remote applications, ensure that the Microsoft Azure region you select for the pod provides for those NV-series VM types that you want to use and which are supported in this Horizon Cloud release. See the Microsoft documentation at <a href="https://azure.microsoft.com/en-us/regions/services/">https://azure.microsoft.com/en-us/regions/services/</a> for details.</td>
</tr>
<tr>
<td>Description</td>
<td>Optional: Enter a description for this pod.</td>
</tr>
<tr>
<td>High Availability</td>
<td>Enable this toggle to deploy a pod that is configured with high availability. For details about high availability and the pod, see High Availability and Your Horizon Cloud Pod in Microsoft Azure in the Administration Guide. If you disable this toggle, the pod is deployed without high availability enabled. Note Even when this toggle is disabled, the pod still deploys with the pod architecture that has the memory-optimized, Gen 5, Microsoft Azure Database for PostgreSQL server and a Microsoft Azure load balancer in front of the pod manager VM. When you see this toggle displayed in the pod deployment wizard, its presence indicates the pod deployer will deploy a pod using the pod architecture that has those items. A pod deployed with this toggle enabled will have a second pod manager VM, while a pod deployed with this toggle disabled will have a single pod manager VM.</td>
</tr>
<tr>
<td>Virtual Network</td>
<td>Select a virtual network from the list. Only virtual networks (VNets) that exist in the region selected in the Microsoft Azure Region field are shown here. You must have already created the VNet you want to use in that region in your Microsoft Azure subscription.</td>
</tr>
<tr>
<td>Use Existing Subnet</td>
<td>Enable this toggle if you have created subnets in advance to meet the pod's subnet requirements. When this toggle is set to Yes, the wizard's fields for specifying subnets change to drop-down selection menus. Important The wizard does not support using an existing subnet for one of the required subnets and also entering CIDR addresses for the other required subnets. When this toggle is set to Yes, you must select from existing subnets for all of the pod's required subnets.</td>
</tr>
<tr>
<td>Option</td>
<td>Description</td>
</tr>
<tr>
<td>--------------------------------</td>
<td>-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>Management Subnet</strong></td>
<td>When <strong>Use Existing Subnet</strong> is enabled, this menu lists the subnets available on the VNet selected for <strong>Virtual Network</strong>. Select the existing subnet that you want to use for the pod’s management subnet.</td>
</tr>
<tr>
<td><strong>Management Subnet (CIDR)</strong></td>
<td><em>Important</em> Select a subnet that has the Microsoft.SQL service configured as a service endpoint for that subnet. This service endpoint supports the required communication between the pod manager VMs and the pod’s Azure Postgres database over the management subnet.</td>
</tr>
<tr>
<td></td>
<td>Select an empty subnet, one that has no other resources attached to it. If the subnet is not empty, unexpected results might occur during the deployment process or pod operations.</td>
</tr>
<tr>
<td></td>
<td>When <strong>Use Existing Subnet</strong> is disabled, enter a subnet address range (in CIDR notation) for the deployer to create a subnet to which the pod and Unified Access Gateway instances will get connected, such as 192.168.8.0/27. For the management subnet, a CIDR of at least /27 is required.</td>
</tr>
<tr>
<td></td>
<td><strong>Caution</strong> When you do not select the wizard option to use existing subnets, the subnet must not already exist in your Microsoft Azure environment. If it already exists, you will get an error when you try to proceed to the next wizard step.</td>
</tr>
<tr>
<td><strong>VM Subnet - Primary</strong></td>
<td>This field relates to the subnet used for those VMs that the pod provisions to provide your end-user desktops and applications. Such VMs include the golden image VMs, the farms’ RDSH-capable VMs, and the VDI desktop VMs.</td>
</tr>
<tr>
<td><strong>VM Subnet (CIDR) - Primary</strong></td>
<td>When <strong>Use Existing Subnet</strong> is enabled, this menu lists the subnets available on the VNet selected for <strong>Virtual Network</strong>. Select the existing subnet that you want used for those VMs.</td>
</tr>
<tr>
<td></td>
<td><strong>Important</strong> Select an empty subnet, one that has no other resources attached to it. If the subnet is not empty, unexpected results might occur during the deployment process or pod operations.</td>
</tr>
<tr>
<td></td>
<td>When <strong>Use Existing Subnet</strong> is disabled, enter a subnet address range (in CIDR notation) for the deployer to create this subnet as the pod is deployed, such as 192.168.12.0/22. For the desktop subnet, a CIDR of at least /27 is required, and a CIDR of /22 is recommended.</td>
</tr>
<tr>
<td></td>
<td><strong>Important</strong> Ensure the range you enter is large enough to allow for accommodating the number of VMs you anticipate you will want this pod to provision to provide your farms’ RDSH-capable VMs and the VDI desktop VMs for your end users. This desktop subnet cannot be extended after the pod is deployed.</td>
</tr>
<tr>
<td></td>
<td><strong>Caution</strong> When you do not select the wizard option to use existing subnets, the subnet must not already exist in your Microsoft Azure environment. If it already exists, you will get an error when you try to proceed to the next wizard step.</td>
</tr>
</tbody>
</table>
Option | Description
--- | ---
NTP Servers | Enter the list of NTP servers you want to use for time synchronization, separated by commas. An NTP server you enter here can be a public NTP server or your own NTP server that you set up for providing time synchronization. The NTP servers you specify here must be reachable from the virtual network you selected in the Virtual Network field for the pod to use. In this field, you can specify each NTP server either by its numeric IP address or its domain name. When you provide a domain name in this field instead of a numeric IP address, you must ensure that the DNS configured for your virtual network can resolve the specified name. Examples of public NTP server domain names are time.windows.com, us.pool.ntp.org, time.google.com.

Use Proxy | If you require a proxy for outbound Internet connectivity, enable this toggle and complete the associated displayed fields. The pod deployer requires outbound access to the Internet to securely download software into the Microsoft Azure cloud environment and connect back to the Horizon Cloud cloud control plane. To enable the pod to use your proxy configuration, you must provide the following information after enabling the toggle.

■ **Proxy** (required): Type the hostname or IP address for your proxy server.

■ **Port** (required): Type the port number that is specified in your proxy server configuration.

If your proxy server configuration requires a user name and password for authentication, provide those credentials also.

![Use Proxy](image)

The following screenshot is an example with this step completed when having the deployment process automatically create the subnets and with **High Availability** enabled. In this example, a proxy was not needed to meet the outbound Internet connectivity requirement.
2 Proceed to the next step by clicking **Next**.

3 Specify details for the pod to have a Unified Access Gateway configuration by following the steps in **Specify the Horizon Cloud Pod's Gateway Configuration**. For the ability to have your end users access their desktops and remote applications over the Internet, an external Unified Access Gateway configuration is required.

**Specify the Horizon Cloud Pod's Gateway Configuration**

In this step of the wizard, specify the information required to deploy the pod with a gateway configured. Unified Access Gateway provides the gateway environment for a pod deployed into
Microsoft Azure. When deploying the new pod, you can choose to have either an external or internal gateway configuration — or have both types on the same pod. By default, when this wizard step displays, the external gateway configuration is selected.

**External gateway configuration**

The external gateway configuration gives the ability to provide access to desktops and applications for end users located outside of your corporate network. When the pod has this external gateway configuration, the pod includes an Azure Load Balancer resource and Unified Access Gateway instances to provide this access. In this case, the instances have three NICs each: one NIC on the management subnet, one NIC on the desktop subnet, and one NIC on the DMZ subnet. In the deployment wizard, you have the option to specify the load balancing type as either private or public, depending on whether you want a private IP or public IP address for the load balancer. If you disable the public IP toggle, then you must specify the IP address that you have mapped in your DNS server to the FQDN that your end users' Horizon clients will use for PCoIP connections to the gateway.

For an external gateway configuration, you also have the option to deploy the configuration into a VNet that is separate from the pod's VNet. The VNets must be peered. This type of configuration gives the ability to deploy the pod into more complex network topologies in Microsoft Azure, such as a hub-spoke network topology.

**Note**  If you enabled the toggle for having the external gateway using its own subscription in the first wizard step, you must deploy the external gateway into its own VNet, the VNet that is associated with that subscription. If you enabled that toggle, you can optionally select an existing resource group in that subscription for the external gateway's resources. You must have prepared that resource group in advance so that you can select it in this wizard step.

**Internal gateway configuration**

The internal gateway configuration gives the ability for end users located inside your corporate network to have trusted HTML Access (Blast) connections to their desktops and applications. If the pod is not configured with this internal gateway configuration, end users inside your corporate network see the standard browser untrusted certificate error when they use their browsers to make HTML Access (Blast) connections to their desktops and applications. When the pod has this internal gateway configuration, the pod includes an Azure Load Balancer resource and Unified Access Gateway instances to provide this access. In this case, the instances have two NICs each: one NIC on the management subnet and one NIC on the desktop subnet. By default, this gateway's load balancing type is private.

The following screenshot is an example of the step when it is initially displayed. Some controls are displayed only when you selected at the first wizard step to use a different subscription for the external gateway configuration.
Prerequisites

Verify that you have met the prerequisites described in Prerequisites for Running the Pod Deployment Wizard.

**Important** To complete this step, you must have the required fully qualified domain name (FQDN) which your end users will use to access the service and have a signed SSL certificate (in PEM format) based on that FQDN. The certificate must be signed by a trusted CA. A single PEM file must contain the entire certificate chain and the private key: SSL certificate intermediate certificates, root CA certificate, private key. For details, see Convert a Certificate File to the PEM Format Required for Pod Deployment.

Verify that all certificates in the certificate chain have valid time frames. If any certificate in the chain is expired, unexpected failures can occur later in the pod onboarding process.

This FQDN cannot contain underscores. In this release, connections to the Unified Access Gateway instances will fail when the FQDN contains underscores.

Procedure

1. If you want the external gateway configuration, complete the fields in the **External Gateway** section.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enable External Gateway?</td>
<td>Controls whether the pod has an external gateway configuration. The external configuration allows access to desktops and applications for users located outside of your corporate network. The pod includes a Microsoft Azure load balancer resource and Unified Access Gateway instances to provide this access.</td>
</tr>
<tr>
<td></td>
<td><strong>Note</strong> Leaving the default enabled setting is recommended.</td>
</tr>
<tr>
<td></td>
<td>When this toggle is switched off, clients must either connect through Workspace ONE Access integrated with the pod or directly to the pod managers' load balancer, or they connect through an internal gateway configuration. In the case of clients connecting through Workspace ONE Access integrated with the pod or directly, some post-deployment steps are required. In this case, after the pod is deployed, see the information in the Horizon Cloud Administration Guide about uploading SSL certificates to the pod.</td>
</tr>
<tr>
<td>FQDN</td>
<td>Enter the required fully qualified domain name (FQDN), such as our0rg.example.com, which your end users will use to access the service. You must own that domain name and have a certificate in PEM format that can validate that FQDN.</td>
</tr>
<tr>
<td></td>
<td><strong>Important</strong> This FQDN cannot contain underscores. In this release, connections to the Unified Access Gateway instances will fail when the FQDN contains underscores.</td>
</tr>
<tr>
<td>DNS Addresses</td>
<td>Optionally enter addresses for additional DNS servers that Unified Access Gateway can use for name resolution, separated by commas. When configuring this external Unified Access Gateway configuration to use two-factor authentication with your on-premises RADIUS server, you would specify the address of a DNS server that can resolve the name of your on-premises RADIUS server. As described in the Prerequisites for All Deployments, a DNS server must be set up internally in your subscription and configured to provide external name resolution. The Unified Access Gateway instances use that DNS server by default. If you specify addresses in this field, the deployed Unified Access Gateway instances use the addresses in addition to the prerequisite DNS server that you configured in your subscription's virtual network.</td>
</tr>
<tr>
<td>Option</td>
<td>Description</td>
</tr>
<tr>
<td>-------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>Routes</strong></td>
<td>Optionally specify custom routes to additional gateways that you want the deployed Unified Access Gateway instances to use to resolve network routing for the end user access. The specified routes are used to allow Unified Access Gateway to resolve network routing such as to RADIUS servers for two-factor authentication. When configuring this pod to use two-factor authentication with an on-premises RADIUS server, you must enter the correct route the Unified Access Gateway instances can use to reach the RADIUS server. For example, if your on-premises RADIUS server uses 10.10.60.20 as its IP address, you would enter 10.10.60.0/24 and your default route gateway address as a custom route. You obtain your default route gateway address from the Express Route or VPN configuration you are using for this environment. Specify the custom routes as a comma-separated list in the form ipv4-network-address/bits ipv4-gateway-address, for example: 192.168.1.0/24 192.168.0.1, 192.168.2.0/24 192.168.0.2.</td>
</tr>
<tr>
<td><strong>VM Model</strong></td>
<td>Select a model to use for the Unified Access Gateway instances. You must ensure that the Microsoft Azure subscription you specified for this pod can provide the capacity for two VMs of the selected model.</td>
</tr>
<tr>
<td><strong>Certificate</strong></td>
<td>Upload the certificate in PEM format that Unified Access Gateway will use to allow clients to trust connections to the Unified Access Gateway instances running in Microsoft Azure. The certificate must be based on the FQDN you entered and be signed by a trusted CA. The PEM file must contain the entire certificate chain and the private key: SSL certificate intermediate certificates, root CA certificate, private key.</td>
</tr>
</tbody>
</table>

Specify the settings for this gateway’s Microsoft Load Balancer.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Enable Public IP?</strong></td>
<td>Controls whether this gateway’s load balancing type is configured as private or public. If switched on, the deployed Microsoft Azure load balancer resource is configured with a public IP address. If switched off, the Microsoft Azure load balancer resource is configured with a private IP address.</td>
</tr>
<tr>
<td><strong>Important</strong></td>
<td>In this release, you cannot later change the external gateway’s load balancing type from public to private, or from private to public. The only way to make that change would be to delete the gateway configuration entirely from the deployed pod and then edit the pod to add it back with the opposite setting.</td>
</tr>
<tr>
<td></td>
<td>If you disable this toggle, the field Public IP for Horizon FQDN appears.</td>
</tr>
<tr>
<td><strong>Public IP for Horizon FQDN</strong></td>
<td>When you have chosen not to configure the deployed Microsoft Azure load balancer with a public IP, you must provide the IP address that you are mapping in your DNS to the FQDN that your end users’ Horizon clients will use for PCoIP connections to the gateway. The deployer will configure this IP address in the Unified Access Gateway configuration settings.</td>
</tr>
</tbody>
</table>

Specify the external gateway’s networking settings.
<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Use a Different Virtual Network</strong></td>
<td>This toggle controls whether the external gateway will be deployed into its own VNet, separate from the pod's VNet. The following rows describe the different cases.</td>
</tr>
<tr>
<td><strong>Note</strong></td>
<td>When you specified to use a different subscription for the external gateway in the first step of the wizard, this toggle is enabled by default. You must choose a VNet for the gateway in that situation.</td>
</tr>
<tr>
<td><strong>Use a Different Virtual Network</strong></td>
<td>When the toggle is disabled, the external gateway will be deployed into the pod's VNet. In this case, you must specify the DMZ subnet.</td>
</tr>
<tr>
<td>DMZ Subnet</td>
<td>When Use Existing Subnet is enabled in the Pod Setup wizard step, DMZ Subnet lists the subnets available on the VNet selected for Virtual Network. Select the existing subnet that you want to use for the pod's DMZ subnet.</td>
</tr>
<tr>
<td><strong>Important</strong></td>
<td>Select an empty subnet, one that has no other resources attached to it. If the subnet is not empty, unexpected results might occur during the deployment process or pod operations.</td>
</tr>
<tr>
<td>DMZ Subnet (CIDR)</td>
<td>When Use Existing Subnet is disabled in the preceding wizard step, enter the subnet (in CIDR notation) for the DMZ (demilitarized zone) network that will be configured to connect the Unified Access Gateway instances to the gateway's Microsoft Azure public load balancer.</td>
</tr>
<tr>
<td><strong>Use a Different Virtual Network</strong></td>
<td>When the toggle is enabled, the external gateway will be deployed into its own VNet. In this case, you must select the VNet to use and then specify the three required subnets. Enable the Use Existing Subnet toggle to select from subnets that you have created in advance on the specified VNet. Otherwise, specify the subnets in CIDR notation.</td>
</tr>
<tr>
<td><strong>Important</strong></td>
<td>Select empty subnets, ones that have no other resources attached to them. If the subnets are not empty, unexpected results might occur during the deployment process or pod operations.</td>
</tr>
<tr>
<td>Management subnet - Specify the subnet to use for the gateway's management subnet. A CIDR of at least /27 is required. This subnet must have the Microsoft.SQL service configured as a service endpoint.</td>
<td></td>
</tr>
<tr>
<td>Back-end subnet - Specify the subnet to use for the gateway's back end subnet. A CIDR of at least /27 is required.</td>
<td></td>
</tr>
<tr>
<td>Front-end subnet - Specify the subnet for the front-end subnet that will be configured to connect the Unified Access Gateway instances to the gateway's Microsoft Azure public load balancer.</td>
<td></td>
</tr>
</tbody>
</table>

2 (Optional) In the **External Gateway** section, optionally configure two-factor authentication for the external gateway.

Complete the steps in **Specify Two-Factor Authentication Capability for the Pod**.
3  (Optional) In the **Deployment** section, use the toggle to optionally select an existing resource group into which you want the deployer to deploy the resources for the external gateway configuration.

This toggle displays when you have specified to use a different subscription for the external gateway in the first step of the wizard. When you enable the toggle, a field appears in which you can search for and select the resource group.

4  In the **Internal Gateway** section, if you want the internal gateway configuration, switch on the **Enable Internal Gateway?** toggle and complete the fields that appear.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enable Internal Gateway?</td>
<td>Controls whether the pod has an internal gateway configuration. The internal configuration provides trusted access to desktops and applications for HTML Access (Blast) connections for users located inside of your corporate network. The pod includes an Azure load balancer resource and Unified Access Gateway instances to provide this access. By default, this gateway's load balancing type is private. The load balancer is configured with a private IP address.</td>
</tr>
<tr>
<td>FQDN</td>
<td>Enter the required fully qualified domain name (FQDN), such as ourOrg.example.com, which your end users will use to access the service. You must own that domain name and have a certificate in PEM format that can validate that FQDN. If you specified an FQDN for the external gateway, you must enter the same FQDN here. <strong>Important</strong> This FQDN cannot contain underscores. In this release, connections to the Unified Access Gateway instances will fail when the FQDN contains underscores.</td>
</tr>
<tr>
<td>DNS Addresses</td>
<td>Optionally enter addresses for additional DNS servers that Unified Access Gateway can use for name resolution, separated by commas. When configuring this internal gateway configuration to use two-factor authentication with your on-premises RADIUS server, you would specify the address of a DNS server that can resolve the name of your on-premises RADIUS server. As described in the <strong>Prerequisites for All Deployments</strong>, a DNS server must be set up internally in your subscription and configured to provide name resolution. The Unified Access Gateway instances use that DNS server by default. If you specify addresses in this field, the deployed Unified Access Gateway instances use the addresses in addition to the prerequisite DNS server that you configured in your subscription's virtual network.</td>
</tr>
<tr>
<td>Routes</td>
<td>Optionally specify custom routes to additional gateways that you want the deployed Unified Access Gateway instances to use to resolve network routing for the end user access. The specified routes are used to allow Unified Access Gateway to resolve network routing such as to RADIUS servers for two-factor authentication. When configuring this pod to use two-factor authentication with an on-premises RADIUS server, you must enter the correct route the Unified Access Gateway instances can use to reach the RADIUS server. For example, if your on-premises RADIUS server uses 10.10.60.20 as its IP address, you would enter 10.10.60.0/24 and your default route gateway address as a custom route. You obtain your default route gateway address from the Express Route or VPN configuration you are using for this environment. Specify the custom routes as a comma-separated list in the form ipv4-network-address/bits ipv4-gateway-address, for example: 192.168.1.0/24 192.168.0.1, 192.168.2.0/24 192.168.0.2.</td>
</tr>
<tr>
<td>Option</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>----------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>VM Model</td>
<td>Select a model to use for the Unified Access Gateway instances. You must ensure that the Microsoft Azure subscription you specified for this pod can provide the capacity for two VMs of the selected model.</td>
</tr>
<tr>
<td>Certificate</td>
<td>Upload the certificate in PEM format that Unified Access Gateway will use to allow clients to trust connections to the Unified Access Gateway instances running in Microsoft Azure. The certificate must be based on the FQDN you entered and be signed by a trusted CA. The PEM file must contain the entire certificate chain and the private key: SSL certificate intermediate certificates, root CA certificate, private key.</td>
</tr>
</tbody>
</table>

5. (Optional) In the **Internal Gateway** section, optionally configure two-factor authentication for the internal Unified Access Gateway.

   Complete the steps in Specify Two-Factor Authentication Capability for the Pod.

**Results**

When you have provided the required information associated with your selected options, you can click **Validate & Proceed** to continue to the wizard's final step. See **Validate and Proceed, and then Start the Pod Deployment Process**.

**Specify Two-Factor Authentication Capability for the Pod**

In the pod deployment wizard step for specifying its Unified Access Gateway configurations, you can also specify use of two-factor authentication for your end users’ access to their desktops and applications through those gateway configurations. You can specify these two-factor authentication details after providing the Unified Access Gateway configuration details.

**Prerequisites**

Verify that you have met the prerequisites described in **Prerequisites for Running the Pod Deployment Wizard**.

For the external or internal Unified Access Gateway configuration for which you are entering the two-factor authentication details, verify that you have completed the fields for the Unified Access Gateway configuration in the wizard as described in **Specify the Horizon Cloud Pod’s Gateway Configuration**. When configuring two-factor authentication to an on-premises authentication server, you also provide information in the following fields so that the Unified Access Gateway instances can resolve routing to that on-premises server.
<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DNS Addresses</td>
<td>Specify one or more addresses of DNS servers that can resolve the name of your on-premises authentication server.</td>
</tr>
<tr>
<td>Routes</td>
<td>Specify one or more custom routes that allow the pod’s Unified Access Gateway instances to resolve network routing to your on-premises authentication server. For example, if you have an on-premises RADIUS server that uses 10.10.60.20 as its IP address, you would use 10.10.60.0/24 and your default route gateway address as a custom route. You obtain your default route gateway address from the Express Route or VPN configuration you are using for this environment. Specify the custom routes as a comma-separated list in the form ipv4-network-address/bits ipv4-gateway-address, for example: 192.168.1.0/24 192.168.0.1, 192.168.2.0/24 192.168.0.2.</td>
</tr>
</tbody>
</table>

Verify that you have the following information used in your authentication server's configuration, so that you can provide it in the appropriate fields in the pod deployment wizard. If you have both a primary and secondary server, obtain the information for each of them.

- IP address or DNS name of the authentication server
- The shared secret that is used for encryption and decryption in the authentication server's protocol messages
- Authentication port numbers, typically the 1812 UDP port.
- Authentication protocol type. The authentication types include PAP (Password Authentication Protocol), CHAP (Challenge Handshake Authentication Protocol), MSCHAP1, MSCHAP2 (Microsoft Challenge Handshake Authentication Protocol, version 1 and 2).

**Note** Check your RADIUS vendor’s documentation for the authentication protocol that your RADIUS vendor recommends and follow their indicated protocol type. The pod’s capability to support two-factor authentication with RADIUS is provided by the Unified Access Gateway instances, and Unified Access Gateway supports PAP, CHAP, MSCHAP1, and MSCHAP2. PAP is generally less secure than MSCHAP2. PAP is also a simpler protocol than MSCHAP2. As a result, even though most RADIUS vendors are compatible with the simpler PAP protocol, some RADIUS vendors are not as compatible with the more secure MSCHAP2.

**Procedure**

1. Switch on the **Enable 2 Factor Authentication** toggle.

When the toggle is enabled, the wizard displays the additional configuration fields. Use the scroll bar to access all of the fields.

The following screenshot is an example of what is displayed after you switch on the toggle in the **External UAG** section.
2 Select your two-factor authentication method in the drop-down list. In this release, RADIUS authentication is supported.

3 In the **Name** field, enter an identifying name for this configuration.

4 In the **Properties** section, specify details related to the end users' interaction with the login screen they will use to authenticate for access.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Display Name</strong></td>
<td>You can leave this field blank. Even though this field is visible in the wizard, it only sets an internal name in Unified Access Gateway. This name is not used by Horizon clients.</td>
</tr>
<tr>
<td><strong>Display Hint</strong></td>
<td>Optionally enter a text string that will be displayed to the end users in the message on the end-user client login screen when it prompts the user for their RADIUS user name and passcode. The specified hint appears to the end user as <em>Enter your DisplayHint user name and passcode</em>, where <em>DisplayHint</em> is the text you specify in this field. This hint can help guide users to enter the correct RADIUS passcode. As an example, specifying a phrase like <em>Example Company user name and domain password below</em> for would result in a prompt to the end user that says <em>Enter your Example Company user name and domain password below</em> for user name and passcode.</td>
</tr>
<tr>
<td><strong>Name ID Suffix</strong></td>
<td>This setting is used in SAML scenarios, where your pod is configured to use TrueSSO for single sign-on. Optionally provide a string which the system will append to the SAML assertion user name that is sent to the broker. For example, if the user name is entered as user1 on the login screen and a name ID suffix of example.com was specified here, the system sends a SAML assertion user name of <a href="mailto:user1@example.com">user1@example.com</a> to the broker.</td>
</tr>
</tbody>
</table>
## Option Description

**Number of Iterations** Enter the maximum number of failed authentication attempts that a user is allowed when attempting to log in using this RADIUS system.

**Maintain Username** Enable this toggle to maintain the user's RADIUS username during authentication to Horizon Cloud. When enabled:
- The user must have the same username credentials for RADIUS as for their Active Directory authentication to Horizon Cloud.
- The user cannot change the username in the login screen.

If this toggle is disabled, the user is able to type a different user name in the login screen.

**Note** For the relationship between enabling Maintain Username and the domain security settings in Horizon Cloud, see the Domain Security Settings on General Settings Page topic in the Horizon Cloud Administration Guide.

5 In the Primary Server section, specify details about the authentication server.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Host Name / IP Address</strong></td>
<td>Enter the DNS name or the IP address of the authentication server.</td>
</tr>
<tr>
<td><strong>Shared Secret</strong></td>
<td>Enter the secret for communicating with the authentication server. The value must be identical to the server-configured value.</td>
</tr>
<tr>
<td><strong>Authentication Port</strong></td>
<td>Specify the UDP port configured on the authentication server for sending or receiving authentication traffic. The default is 1812.</td>
</tr>
<tr>
<td><strong>Accounting Port</strong></td>
<td>Optionally specify the UDP port configured on the authentication server for sending or receiving accounting traffic. The default is 1813.</td>
</tr>
<tr>
<td><strong>Mechanism</strong></td>
<td>Select the authentication protocol that is supported by the specified authentication server and which you want the deployed pod to use.</td>
</tr>
<tr>
<td><strong>Server Timeout</strong></td>
<td>Specify the number of seconds that the pod should wait for a response from the authentication server. After this number of seconds, a retry is sent if the server does not respond.</td>
</tr>
<tr>
<td><strong>Max Number of Retries</strong></td>
<td>Specify the maximum number of times the pod should retry failed requests to the authentication server.</td>
</tr>
<tr>
<td><strong>Realm Prefix</strong></td>
<td>Optionally provide a string which the system will place at the beginning of the user name when the name is sent to the authentication server. The user account location is called the realm. For example, if the user name is entered as user1 on the login screen and a realm prefix of DOMAIN-A\ was specified here, the system sends DOMAIN-A\user1 to the authentication server. If you do not specify a realm prefix, only the entered user name is sent.</td>
</tr>
<tr>
<td><strong>Realm Suffix</strong></td>
<td>Optionally provide a string which the system will append to the user name when the name is sent to the authentication server. For example, if the user name is entered as user1 on the login screen and a realm suffix of @example.com was specified here, the system sends <a href="mailto:user1@example.com">user1@example.com</a> to the authentication server.</td>
</tr>
</tbody>
</table>
6 (Optional) In the Secondary Server section, optionally specify details about an auxiliary authentication server.

You can configure a secondary authentication server to provide for high availability. Enable the **Auxiliary Server** toggle and complete the fields as described in Step Primary Server section.

**Validate and Proceed, and then Start the Pod Deployment Process**

After you click **Validate & Proceed**, the system verifies your specified values. If everything validates, the wizard displays a summary of the information for your review. Then you start the deployment process.

**Procedure**

1 Click **Validate & Proceed**.

The system validates your specified values, such as:

- Are the specified address ranges for the to-be-created subnets valid and non-overlapping with other addresses in the selected region within your subscription.
- Are there enough virtual machine (VM) and cores in your subscription's quota to build out the pod.
- Are any uploaded certificate files in the correct PEM format.
- If you selected to use an existing management subnet, does it have the Microsoft.SQL service endpoint enabled on that subnet?

**Important** Starting with the September 2019 service release, new pod deployments require the Microsoft.SQL service endpoint enabled on the management subnet to support use of the pod's Microsoft Azure PostgreSQL database. If you see a validation error similar to the following screenshot that lists your selected subnet, it means the pre-existing management subnet that you selected in the wizard is missing that Microsoft.SQL service endpoint configured on it. At this point, you can log in to the Microsoft Azure portal and enable the Microsoft.SQL service endpoint on the subnet. Then you can resubmit the wizard to deploy the pod. For some details on how to enable that endpoint, see In Advance of Pod Deployment, Create the Horizon Cloud Pod's Required Subnets on your VNet in Microsoft Azure.

If everything validates, the summary page displays.
If you see an error message about overlapping network addresses, verify whether you have existing subnets using the same values already in your subscription.

2 On the final wizard step, review the summarized information and click **Submit**.

The system starts deploying the pod into your Microsoft Azure environment.

![Microsoft Azure, 0 Pod](image)

**Results**

Deploying your first pod can take up to an hour. Until the pod is successfully deployed, a progress icon is displayed in the console's Getting Started screen. You might need to refresh the screen in your browser to see the progress. The browser-based user interface can time out after approximately 30 minutes and ask you to log back in.

**Important** The pod's pending stage typically lasts up to ten minutes. However, when deploying a pod in Microsoft Azure China cloud, the overall deployment process can take up to seven (7) hours to complete. The process is subject to geographic network issues that can cause slow download speeds as the binaries are downloaded from the cloud control plane.

If the pod has not moved from Pending to Downloading state after 20 minutes, and you are not deploying into Microsoft Azure China, the system automatically puts the pod into Error state and displays a message that states the pod cannot connect to the cloud services and to check the networking connectivity in your Microsoft Azure environment.

If the display shows the pod is in Error state because the deployed jump box VM could not download the binaries it needs from the cloud service, there is likely an issue with your environment's network configuration. For example, the VNet's configured DNS might not be resolving internal or external names or the required outbound ports are not open or are blocked by your firewall. Sometimes there is a temporary loss of connectivity to packages.microsoft.com site used to download the Microsoft Azure Command Line Interface software. You can run some tests to verify if your environment's networking is configured properly for the pod's requirements. See [Troubleshooting If You Encounter Pod Deployment or First-Time Domain Bind Issues](#).

Throughout the pod deployment process, the Getting Started page's Capacity section indicates the various stages the process goes through (pending, downloading, building, connecting, and so on).
The following table gives some approximate sample durations for the stages in building the pod.

**Important** The actual durations you experience in your deployment's progress will vary depending on the network latencies that exist at the time.

<table>
<thead>
<tr>
<th>Stage</th>
<th>Sample duration</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pending</td>
<td>10 minutes</td>
</tr>
<tr>
<td>Downloading</td>
<td>10 minutes</td>
</tr>
<tr>
<td>Building</td>
<td>15 minutes</td>
</tr>
<tr>
<td>Connecting</td>
<td>10 minutes</td>
</tr>
</tbody>
</table>

When the pod is successfully deployed:

- Horizon Cloud sends a notification email to the account owner that is identified in the corresponding Horizon Cloud customer account record. The email states the pod onboarding is complete.

- A green checkmark is displayed in the Getting Started screen next to a message saying the pod was added and about completing the domain join process.

At this point, because your Active Directory domain is not yet registered with the pod, a **Delete Pod** option is available in the **Manage** menu. If the deployment process fails for some reason or if you dislike the values you used and want to start over before registering your Active Directory domain, you can click **Manage > Delete Pod** to delete the artifacts that were deployed. When the screen indicates the pod is successfully deleted, you can start the process over by clicking **Manage > Add Pod** again. The following screenshot illustrates the location of the **Manage > Delete Pod** option.
If you choose to delete the pod from this point, due to network latency, the Getting Started page might indicate the pod is fully deleted before all of the pod-related artifacts are completely deleted from your Microsoft Azure environment. Before running the pod deployment wizard again after deleting the new pod, take the following steps:

1. Log out of the Horizon Cloud user interface.
2. Log in to the Microsoft Azure portal.
3. Navigate to your VNet.
4. If you had the deployer automatically create the pod's subnets, verify that no pod-created subnets exist and that the address ranges that you specified for the pod's subnets have been removed from the VNet's address space.

Then you can log back in to Horizon Cloud to run the pod deployment wizard again.

**What to do next**

Expand the General Setup section of the Getting Started screen and complete the required task of registering an Active Directory domain. Registering Active Directory is the next required step. After registering the domain and setting the Super Administrator role for a domain group, the system makes all of the console accessible to you. Then you continue management of this pod in the console. See the Getting Started chapter of the Horizon Cloud Administration Guide. After registering the Active Directory domain, follow the Getting Started wizard to see which task to complete next.

You must set up the appropriate CNAME records in your DNS server according to the type of gateway you specified.

- For an external gateway enabled with a public IP address, map the FQDN that you entered in the deployment wizard to the gateway's Microsoft Azure load balancer resource's auto-generated public FQDN. Your DNS server record maps that Microsoft Azure load balancer's auto-generated public FQDN with the FQDN that your end users will use, and which is used in the uploaded certificate. The following code line demonstrates an example. You locate the ID to use from the pod's details page in the console, after you have registered the Active Directory domain. If the external gateway was deployed in its own VNet, use the ID that is displayed in the Deployment ID field.

```
ourApps.ourOrg.example.com   vwm-hcs-ID-uag.region.cloudapp.azure.com
```

- For an internal gateway or an external gateway without a public IP address, map the FQDN that you entered in the deployment wizard to the gateway's Microsoft Azure load balancer resource's private IP address. Your DNS server record maps that Microsoft Azure load balancer's IP address with the FQDN that your end users will use, and which is used in the uploaded certificate. The following code line demonstrates an example.

```
ourApps.ourOrg.example.com   Azure-load-balancer-private-IP
```
When both the external and internal gateway configurations use the same FQDN, after the pod is deployed you must configure the routing of the incoming end-user client traffic to the appropriate load balancer resource in the gateways' resource groups. The goal is to set up the routing so that client traffic from the Internet is routed to the external gateway's Microsoft Azure Public Load Balancer and client traffic from your intranet is routed to the internal gateway's Microsoft Azure Internal Load Balancer. When both gateways have the same FQDN, you configure Split DNS (Split Domain Name System) to resolve the gateway address either to the external gateway or internal gateway depending on the origin network of the end-user client's DNS query.

See the *Horizon Cloud Administration Guide* for the steps to locate the load balancer information in the pod's details page.

If you specified RADIUS two-factor authentication for the pod's gateway configurations, you must complete the following tasks.

- If you configured an external gateway with RADIUS settings and that RADIUS server is not reachable within the same VNet as used by the pod, or within the peered VNet topology if you deployed the external gateway into its own VNet, verify, configure that RADIUS server to allow client connections from the IP address of the external gateway's load balancer. In an external gateway configuration, the Unified Access Gateway instances attempt contact with the RADIUS server using that load balancer address. To allow the connections, ensure the load balancer resource's IP address that is in that external gateway's resource group is specified as a client in your RADIUS server configuration.

- If you configured an internal gateway, or an external gateway and your RADIUS server is reachable within the same VNet as used by the pod, configure the RADIUS server to allow connections from the appropriate NICs that were created in the gateway's resource group in Microsoft Azure that must communicate with the RADIUS server. Your network administrator determines the RADIUS server's network visibility to the pod's Azure Virtual Network and subnets. Your RADIUS server must allow client connections from the IP addresses of those gateway NICs that correspond to the subnet for which your network administrator has given network visibility to the RADIUS server. The gateway's resource group in Microsoft Azure has four NICs that correspond to that subnet, two that are currently active for the two Unified Access Gateway instances and two that are idle and will become the active ones after the pod goes through an update. To support connectivity between the gateway and the RADIUS server both for ongoing pod operations and after each pod update, ensure the IP addresses of those four NICs are specified as clients in the RADIUS server configuration.

For information on how to obtain those IP addresses, see the *Update Your RADIUS System with the Required Horizon Cloud Pod Gateway Information* topic in the *Horizon Cloud Administration Guide*. 
Troubleshooting If You Encounter Pod Deployment or First-Time Domain Bind Issues

If your environment’s networking is not configured properly for use with the Horizon Cloud pod in Microsoft Azure, the process to build out the pod can get stuck in PENDING state or the post-deployment action to domain bind to your Active Directory environment might fail. The two most common network-related causes are failing to open the required outbound ports and failing to enable the DNS to resolve both internal and external addresses. By following the troubleshooting steps here, you can run some tests to verify the required outbound ports are open and the DNS can resolve both internal and external addresses.

The overall networking requirements for successfully deploying a pod are stated in the prerequisites checklist document, located at this PDF link and described in Configure the DNS Server Settings Needed by the VNet Topology You Will Use for Your Horizon Cloud Pods in Microsoft Azure and DNS Requirements for a Horizon Cloud Pod in Microsoft Azure. If your environment’s networking does not meet those requirements, you will encounter one or both of these two issues:

<table>
<thead>
<tr>
<th>Issues</th>
<th>Common causes</th>
</tr>
</thead>
<tbody>
<tr>
<td>■ The Getting Started page shows the pod in pending state and it never proceeds to connecting state. Usually a pod is in pending state for about 10 minutes (except when deploying a pod into Microsoft Azure China cloud, which takes longer).</td>
<td>■ Required outbound ports are not open, or are blocked by your firewall environment. If the required outbound ports are not open or are blocked by a firewall, it prevents the pod software from securely downloading into the Microsoft Azure cloud environment and connecting back to the Horizon Cloud cloud control plane. As a result, the pending state issue occurs.</td>
</tr>
<tr>
<td>■ Even when the pod has successfully deployed, when you attempt to register your Active Directory, the domain-bind step fails with the error Unable to register Active Directory</td>
<td>■ The VNet DNS server is not properly configured to point to a valid DNS server that can resolve both internal and external machine names.</td>
</tr>
<tr>
<td>■ Though the VNet DNS server is properly pointing to a DNS server, that DNS server cannot resolve both internal and external machine names.</td>
<td>■ Though the VNet DNS server is properly pointing to a DNS server, that DNS server cannot resolve both internal and external machine names. If no DNS resolution for external machine names is provided to the VNet, the pending state issue and domain-bind issue can occur. For example, if the DNS cannot resolve to the Active Directory on the Domain Controllers, the domain-bind step fails. For details about the VNet DNS configuration, see Configure the DNS Server Settings Needed by the VNet Topology You Will Use for Your Horizon Cloud Pods in Microsoft Azure.</td>
</tr>
</tbody>
</table>

To run some tests that will check that the DNS configuration can resolve internal and external names and check that the required outbound ports are open, you deploy a small test virtual machine (VM) into your Microsoft Azure subscription and then use that VM to run these networking tests. The high-level sequence of troubleshooting steps is:

1. Create an SSH key pair.
2. Create the test VM in your Microsoft Azure subscription.
3. Connect to that test VM.
4. Run the networking tests.
When the testing is done, delete the test VM and all the test-related artifacts that were created in your Microsoft Azure environment for doing this troubleshooting.

**Note** If you do not delete the test-related artifacts and you later use the console’s **Delete** action to delete the pod, unexpected results might occur. When deleting a pod, the system checks the pod’s subnets to verify that everything connected to the subnets belongs to the pod itself (according to the pod’s ID). If the system determines additional VMs, VM disks, IPs, or other artifacts are connected to the pod’s subnets, the system cannot cleanly delete the pod.

For details about running the troubleshooting tests, see the following sections.

**Important** If you are directing all traffic out through your on-premises network and only allowing authenticated traffic to pass, but you did not provide values for using a proxy in the pod deployment wizard, even though all of these manual tests will succeed, the traffic sent by an unauthenticated source, the jump box, will fail. The symptom of this situation is the pod deployment is stuck in pending state. If you are in this situation, you must delete the pod from the Getting Started page, re-run the pod deployment wizard, and specify the required proxy information.

**Procedure**

1. **Create an SSH Key Pair**
   You need an SSH key pair to authenticate to the test Linux VM that you will be deploying into your Microsoft Azure subscription. You create the key pair on the system you will use to SSH connect to the test VM. This step is optional if you already have a key pair on that system.

2. **Create the Test Virtual Machine in Your Microsoft Azure Subscription**
   You will use a test Linux virtual machine (VM) in your Microsoft Azure environment to run the tests that check the network connectivity that is configured for your Horizon Cloud pod.

3. **Use SSH to Connect to the Test VM**
   Make a SSH (Secure Shell) connection to the test VM so that you can run the network connectivity tests in your Microsoft Azure environment.

4. **Run the Tests to Check Networking in Your Microsoft Azure Environment**
   You run these tests to verify these two network-related areas are configured properly: that the DNS can resolve both internal and external addresses and that the required outbound ports are open. You run these tests using your test VM.

5. **Delete the Test VM After You Complete the Tests**
   When you have finished the tests to check your Microsoft Azure network configuration and no longer need the test VM, you should delete it and all of its related artifacts from your Microsoft Azure environment.
Create an SSH Key Pair

You need an SSH key pair to authenticate to the test Linux VM that you will be deploying into your Microsoft Azure subscription. You create the key pair on the system you will use to SSH connect to the test VM. This step is optional if you already have a key pair on that system.

You can use either a Microsoft Windows or a Linux system to create this SSH key pair. The steps for both types of systems are described here. Choose the steps most suitable for your situation.

Create an SSH Key Pair on a Microsoft Windows System

Use these steps when you will be using a Microsoft Windows system to SSH connect to the test Linux VM that you will be deploying into your Microsoft Azure subscription.

When you create the test VM in Microsoft Azure, you will use the contents of the generated public key file. If you already have an existing SSH key pair on the Microsoft Windows system that you will use to connect with the test VM, then you can skip this step and proceed with creating the test VM, as described in Create the Test Virtual Machine in Your Microsoft Azure Subscription.

By following these steps, you generate the SSH key pair, copy the public key file's contents so you can use it when creating the test VM, and load the private key into the PuTTY Pageant tool. Pageant is an SSH authentication agent that can hold your private keys in memory. By holding the private key in memory, the private key is automatically applied against any SSH session from that Microsoft Windows system, making it easier to use.

Prerequisites

A Microsoft Windows system does not have SSH key-pair software installed on it by default. Verify that SSH key-pair generating software is installed on the system you are planning to use. You can use any SSH key-pair generating software. The steps below describe using the PuTTY software on Microsoft Windows to create the SSH key pair. You can obtain the PuTTY software from www.putty.org. After the installation, the PuTTY suite of tools is available. The following screenshot shows an example of the PuTTY tools in the Start menu.
Procedure

1. On your Microsoft Windows system, launch PuTTYgen (the PuTTY key generator).

   On Microsoft Windows 10, the PuTTYgen choice in the Start menu looks like

   ![PuTTYgen](image)

   The PuTTY Key Generator window displays. As highlighted in the following screenshot, the goal is to generate a public-private key pair, of type SSH-2 RSA, and having 2048 bits.

   ![PuTTY Key Generator](image)

2. Verify that **SSH-2RSA** is selected, **2048** is set for the number of bits, and then click **Generate**. The window changes to the Key window that displays a progress bar.
3 Move your cursor around randomly in the blank area underneath the progress bar. Moving your cursor around in the area adds required randomness into the process.

![Randomness Image]

4 Save the private key to the system by entering a key passphrase and click **Save private key**.

**Note** Using a key passphrase is an optional best practice. However, if you click **Save private key** without entering a key passphrase, a pop-up window asks you to confirm whether you want to save the private key without a key passphrase.

![PuTTY Key Generator]

The private key is saved as a PPK file. After you click **Save private key**, you can browse to a directory in the local system, type a file name, and save the file.
5 Use the **Save public key** button to save the public key to a location where you can copy it from when you create the test VM.

6 Launch Pageant, the PuTTY SSH authentication agent.

On Microsoft Windows 10, the Pageant choice in the Start menu looks like ![Pageant icon](image). When you click it, the Pageant icon of a computer wearing a hat is loaded into the system tray.

The following screenshot shows the Pageant icon loaded into a Microsoft Windows 10 system tray.

7 Add your private key to Pageant by right-clicking that system tray icon, clicking **Add Key**, and using the file selection window to navigate to and select your saved private key (PPK) file.

![Pageant icon](image)

**Note** If you specified a key passphrase when you saved the private key file earlier, a box is displayed for you to type that passphrase.

**Results**

At this point, the private key is loaded into Pageant. You can use the **View Keys** choice on the action menu to see the key in the list of loaded keys. When you start an SSH session using PuTTY, PuTTY will retrieve the key automatically from Pageant and use the key to authenticate without you having to type your passphrase. Later, when you are finished running SSH sessions
and want to shut down Pageant, use the **Exit** choice from the Pageant system tray icon's right-click menu.

**What to do next**

Create the test VM by following the steps in [Create the Test Virtual Machine in Your Microsoft Azure Subscription](#).

**Create an SSH Key Pair on a Linux System**

Use these steps when you will be using a Linux system to SSH connect to the test Linux VM that you will be deploying into your Microsoft Azure subscription.

In the steps to create the test VM in Microsoft Azure, you will use the contents of the generated public key file. If you already have an existing SSH key pair on the Linux system that you will use to connect with the test VM, then you can skip this step and proceed with creating the test VM, as described in [Create the Test Virtual Machine in Your Microsoft Azure Subscription](#).

**Prerequisites**

Before performing these steps, ensure that you will not overwrite an existing SSH key pair that you want to keep for other purposes. On a Linux system, the SSH public and private key files are created in the Linux `~/.ssh/id_rsa` directory by default. If an SSH key pair exists in that directory and you use the same file name when running this command, or if you specify a different location in the command and an SSH key pair already exists in that location, that existing one is overwritten.

**Procedure**

1. On your Linux system, open a bash shell.
2. In the bash shell, type the following command:
   
   ```bash
   ssh-keygen -t rsa -b 2048
   ```
3. Follow the on-screen instructions about entering a file in which to save the key, entering a passphrase, and confirming the passphrase.

   Here is a sample of the on-screen instructions, where `mykey` was entered as the file in which to save the key.

   ```bash
   $ ssh-keygen -t rsa -b 2048
   Generating public/private rsa key pair.
   Enter file in which to save the key (/mts-cm/home/user1/.ssh/id_rsa): mykey
   Enter passphrase (empty for no passphrase):
   Enter same passphrase again:
   ```

   **Note** Using a key passphrase is an optional best practice.
The private key is saved in the file that you specify and the public key is saved to a file with that same name and a .pub extension. Using the example above of entering mykey as the file, the sample output would be:

```
Your identification has been saved in mykey.
Your public key has been saved in mykey.pub.
```

**What to do next**

Create the test VM by following the steps in Create the Test Virtual Machine in Your Microsoft Azure Subscription.

**Create the Test Virtual Machine in Your Microsoft Azure Subscription**

You will use a test Linux virtual machine (VM) in your Microsoft Azure environment to run the tests that check the network connectivity that is configured for your Horizon Cloud pod.

**Prerequisites**

Verify that you have the SSH public key that you created as described in Create an SSH Key Pair. You will provide that public key in the VM creation wizard so that the VM will trust SSH connections coming from the system that has the corresponding private key.

Verify you have the name of the virtual network (VNet) that is the same one that you are using to deploy your pod, as described in Configure the Required Virtual Network in Microsoft Azure.

If you tried to deploy a pod and the deployment process failed at some point, the process might already have created the pod's management subnet in the VNet.

- If so, it is recommended you deploy the test VM on to that subnet. To identify if the pod's management subnet exists on the VNet, log in to the Microsoft Azure portal, navigate to that VNet, and examine the list of subnets it has. You can navigate to the VNet using the Virtual networks choice in the left navigation bar. The pod's management subnet will have a name in the pattern vmw-hcs-podID-net-management, where podID is the pod's UUID.

- If the pod deployment process did not create the pod's management subnet on the VNet, you can choose any available subnet on the VNet or create a new subnet for the test VM to use.

**Procedure**

1. Log in to the Microsoft Azure portal.

2. From the portal's left navigation bar, click Virtual machines (Virtual machines) and then click + Add.
3 Search for Ubuntu Server 16.04 LTS and select it.

4 Select **Resource Manager** as the deployment model and click **Create**.

The Create Virtual Machine wizard opens to the steps for configuring basic settings.

5 Provide the VM's basic settings and then click **OK** to move to the next wizard step.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Type a name for the VM.</td>
</tr>
<tr>
<td>VM disk type</td>
<td>Keep the default SSD setting.</td>
</tr>
<tr>
<td>User name</td>
<td>Enter a user name that meets the Microsoft Azure user name requirements, as described in the Microsoft documentation <a href="#">here</a>.</td>
</tr>
<tr>
<td></td>
<td><strong>Important</strong> Make a note of this name because you will need to use it later.</td>
</tr>
<tr>
<td>Authentication type</td>
<td>Select <strong>SSH public key</strong>.</td>
</tr>
<tr>
<td>SSH public key</td>
<td>In this field, paste your SSH public key that you created when you created the SSH key pair. The pasted-in contents must start with the line <code>---- BEGIN SSH2 PUBLIC KEY -----</code> and end with the line <code>----- END SSH2 PUBLIC KEY -----</code> from your public key.</td>
</tr>
<tr>
<td>Subscription</td>
<td>Select the same subscription that you are using for your pod.</td>
</tr>
<tr>
<td>Resource group</td>
<td>The recommended choice is to create a new resource group for the test VM and its related artifacts like its disk. Select <strong>Create new</strong> and enter a name for the new resource group. Even though you can use an existing resource group with this test VM, a resource group specific for the test VM is recommended because it is easier to delete the VM and its related artifacts simply by deleting the whole resource group when you are finished running the tests.</td>
</tr>
<tr>
<td>Location</td>
<td>Select the same physical geographic region that you are using for your pod.</td>
</tr>
</tbody>
</table>
6 In the wizard's size step, click a VM size click and then Select to move to the next wizard step.

Because this is expected to be a short-lived VM, used only to complete the verification tests, you can pick any size. However, because smaller sizes usually have lower associated costs in Microsoft Azure, choosing a small size for the test VM is typical. The following screenshot illustrates the example of choosing the D2S_V3 Standard size.
7 In the wizard's settings step, specify the key networking choices for the test VM.

You make three important choices in this wizard step. The following screenshot illustrates these three key items. After setting the three key network options, you can retain all of the other default values.
<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Virtual network</td>
<td>You must select the same VNet that you are using to deploy your pod. This VNet should be the one that you configured according to the details in the prerequisites checklist and as described in Configure the Required Virtual Network in Microsoft Azure.</td>
</tr>
<tr>
<td>Subnet</td>
<td>If you already tried to deploy the pod and the process failed, the pod's management subnet might have been created in the virtual network. If the subnet is there, it is recommended to select that subnet for this test VM. Click the <strong>Subnet</strong> choice to navigate to the subnets that exist on the selected virtual network. You might have to hover over the subnet to see its full name in the tooltip. This screenshot illustrates hovering over a subnet to see the naming pattern of a pod's management subnet, in the pattern <code>vmw-hcs-podID-net-management</code>. If the pod deployment process did not create the pod's management subnet on the VNet, select the subnet on your VNet that you identified to use for the test VM (as described in the prerequisites above). <strong>Note</strong> If the pod was successfully deployed, but you are troubleshooting domain-join issues, you might select the pod's desktop subnet for the test VM instead of the management subnet, because domain-join operations are used with the desktop images that get connected to that desktop subnet.</td>
</tr>
<tr>
<td>Public IP address</td>
<td>Select this choice so that the created test VM will have a public IP address assigned to it. Having a public IP address enables you to connect to it over the wide area network (WAN). <strong>Note</strong> Using a public IP might not be technically feasible in your networking configuration. If you cannot create the test VM with a public IP, you will need to have network connectivity from your local system to the subnet you selected in the <strong>Subnet</strong> field or will need to connect to some other machine on your network and then inbound connect to the test VM.</td>
</tr>
</tbody>
</table>

8. Click **OK** to move to the wizard's summary step.

9. In the summary step, verify that the key pieces of information (subscription, location, virtual network, and subnet) match the ones that you are using for your pod, and then click **Create**.
Results

Creation of the test VM starts running. Typically, you can see the process running on your Microsoft Azure dashboard, as illustrated in the following screenshot.

Deploying the VM usually takes around five to ten minutes. When the VM is fully deployed, it is in Running state. The following screenshot illustrates the details of a sample test VM.

When you see the test VM is up and running, connect to it. Follow the steps in Use SSH to Connect to the Test VM.

Use SSH to Connect to the Test VM

Make a SSH (Secure Shell) connection to the test VM so that you can run the network connectivity tests in your Microsoft Azure environment.

SSH Connect to the Test VM from a Microsoft Windows System

You make this connection from the Microsoft Windows system that has the private key that corresponds to the public key you specified when you created the test VM.

Prerequisites

Verify you have the test VM's IP address and the user name you specified when you created the VM.
On a Microsoft Windows system, PuTTY is typically used. To make it easy for PuTTY to load your private key when you start the SSH session, before starting PuTTY, start Pageant as described in Create an SSH Key Pair on a Microsoft Windows System and add the SSH private key to the Pageant key list. The SSH private key must match the public key you specified when creating the test VM. When the private key is loaded into Pageant, the PuTTY SSH session will use that private key automatically.

**Procedure**

1. Launch PuTTY ( ).
   
   The PuTTY Configuration window opens.

2. In the PuTTY Configuration window, specify the host name, select SSH, and then click Open.
   
   In the PuTTY Configuration window’s Host Name field, type a string in the pattern

   ```plaintext
testvm_username@testvmip
```

   substituting the test VM’s user name and IP address for `testvm_username` and `testvmip` in the string.

   **Important** After you click Open, when this is the first time you connect to the test VM, a PuTTY security message displays stating that the server’s host key is not cached and displays the server’s rsa2 key fingerprint. You can continue to make the connection by either clicking Yes to add the server's host key into PuTTY’s cache or No to connect without adding the key to PuTTY’s cache. If you suspect the connection might not be going to your test VM, click Cancel to abandon the connection and return to the PuTTY Configuration window to verify your host name entry.

   The following screenshot is an illustration of the window using the sample

   ```plaintext
testvmadmin@40.121.180.132
```
Results

When the SSH connection is established, a command-line window displays that looks similar to the following screenshot.
What to do next

Now that you are connected to the test VM, you can run the tests to check network connectivity within your Microsoft Azure environment. Follow the steps described in Run the Tests to Check Networking in Your Microsoft Azure Environment.

SSH Connect to the Test VM from a Linux System

You make this connection from the Linux system that has the private key that corresponds to the public key you specified when you created the test VM.

Prerequisites

Verify you have the test VM's IP address and the user name you specified when you created the VM.

Procedure

1. Open a bash shell.

2. At the bash shell $ prompt, enter the ssh command as below, substituting the test VM's IP address and the user name for testvmip and testvm_username in the command:

   ```bash
   ssh testvm_username@testvmip
   ```

   For example, using the test VM details from the examples in Create the Test Virtual Machine in Your Microsoft Azure Subscription, the sample command would look like:

   ```bash
   ssh testvmadmin@40.121.180.132
   ```

Results

When the SSH connection is established, a command-line window displays that looks similar to the following screenshot.
What to do next

Now that you are connected to the test VM, you can run the tests to check network connectivity within your Microsoft Azure environment. Follow the steps described in Run the Tests to Check Networking in Your Microsoft Azure Environment.

Run the Tests to Check Networking in Your Microsoft Azure Environment

You run these tests to verify these two network-related areas are configured properly: that the DNS can resolve both internal and external addresses and that the required outbound ports are open. You run these tests using your test VM.

The pod relies on DNS to resolve both internal and external addresses. The first two tests here check whether the DNS configured in your network environment can resolve known FQDNs for internal and external addresses.

**Important**  If you are directing all traffic out through your on-premises network and only allowing authenticated traffic to pass, but you did not provide values for using a proxy in the pod deployment wizard, even though all of these manual tests will succeed, the traffic sent by an unauthenticated source, the jump box, will fail. The symptom of this situation is the pod deployment is stuck in pending state. If you are in this situation, you must delete the pod from the Getting Started page, re-run the pod deployment wizard, and specify the required proxy information.

**Prerequisites**

Before running these tests, verify that you created a test VM in your Microsoft Azure subscription and have an SSH connection to it, as described in Create the Test Virtual Machine in Your Microsoft Azure Subscription and Use SSH to Connect to the Test VM.
Obtain the IP addresses and fully qualified domain names (FQDNs) for servers that are internal to your network that you expect to be reachable from the VNet, such as your Active Directory Domain Controller. You will use this information in the DNS verification test.

Procedure

1. Check that DNS is working in your environment to resolve internal FQDNs by using the `dig` command to query a known domain name that is internal to your VNet in Microsoft Azure.

In the SSH connection window, issue the `dig` command to query the domain name of a server that you know is internal to your network, such as your Active Directory Domain Controller.

   ```
dig internal-domain-name
   ```

Where `internal-domain-name` is the fully qualified domain name of a server that you know is internal to your network.

dig (Domain Information Groper) is a command-line tool for network troubleshooting. By running this command using an internal host name, the result verifies that your DNS configuration can resolve internal addresses properly. If your DNS configuration can resolve the `internal-domain-name` used in the command, the command output will return the correct IP address associated with that domain name.

For example, assume the VNet is configured with an internal Active Directory server having Active Directory Domain Controller with a DNS entry of `skylo.local` and an IP address of `192.168.0.15`. Issuing `dig skylo.local` would check whether the VNet's DNS configuration can resolve that internal `skylo.local` server name:

```
testvadmin@HCS-testingVM:~$ dig skylo.local

>; <<>> DiG 9.10.3-P4-Ubuntu <<< skylo.local
;; global options: +cmd
;; Got answer:
;;Arena: QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; Question section:
;skylo.local. IN A

;; Answer section:
skylo.local. 600 IN A 192.168.0.15

;; Query time: 1 msec
;; SERVER: 192.168.0.15#53(192.168.0.15)
;; WHEN: Mon Mar 26 20:58:01 UTC 2018
;; MSG SIZE  rcvd: 56

testvadmin@HCS-testingVM:~$
```
The test is successful when the ANSWER SECTION indicates the provided host name was resolved to the IP address that you expect for that host name.

**Note** Sometimes DNS is not 100% reliable, and some requests resolve fine while others fail. If issuing the command fails the first time, run the command for ten to twenty iterations and see whether you get reliable responses each time.

2 Check that DNS is working in your environment to resolve external FQDNs by using the `dig` command to query a known external domain name.

In the SSH connection window, issue the `dig` command to query an external industry-standard domain name, such as `vmware.com` or `microsoft.com`.

```
dig external-domain-name
```

Where `external-domain-name` is a fully qualified domain name that is external to your VNet. For example, issuing `dig vmware.com` would check whether the VNet's DNS configuration could resolve that external name:

```
testvmadmin@HCS-testingVM:~$ dig vmware.com
; <<>> DiG 9.10.3-P4-Ubuntu <<>> vmware.com
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 38655
;; flags: qr rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
; EDNS: version: 0, flags:; udp: 4000
;; QUESTION SECTION:
;vmware.com.                    IN      A

;; ANSWER SECTION:
vmware.com.             150     IN      A       107.154.105.19
vmware.com.             150     IN      A       107.154.106.19

;; Query time: 28 msec
;; SERVER: 192.168.0.15#53(192.168.0.15)
;; WHEN: Mon Mar 26 21:14:29 UTC 2018
;; MSG SIZE  rcvd: 71

testvmadmin@HCS-testingVM:~$
```

In the above example, the ANSWER SECTION indicates the external domain name `vmware.com` was properly resolved to two IP addresses.

**Note** You can repeat this test using various external domain names, such as `azure.com` or `microsoft.com`, to verify that your DNS can resolve different external names.
If the DNS tests do not work, verify your network configurations and your DNS server. Check that you added your DNS server to your VNet.

**Important** If you find that you need to add your DNS server to your VNet or you have to change the VNet's DNS server configuration, you must restart all VMs that are connected to that VNet for them to pick up the change. If you change the VNet's DNS server configuration and do not restart all of the VMs connected to that VNet, the changes will not propagate correctly on the VNet.

3 Check that the required outbound ports are available by using the `netcat` command.

Horizon Cloud requires some outbound ports to be opened, so that the pod software can be securely downloaded into your Microsoft Azure environment and so that the pod can connect back to the Horizon Cloud control plane. As described in DNS Requirements for a Horizon Cloud Pod in Microsoft Azure, the following outbound TCP ports are required to be open from the pod's management subnet: port 80, 443, and 11371. By running the `netcat` command as indicated in the command below, you can verify that those outbound ports are open as required.

In the SSH connection window, issue the following commands (one per port).

**Note** The command below to test port 11371 specifies `packages.microsoft.com` to test that connection, while the other two lines test the outbound connection to the Horizon Cloud control plane.

```
testvmadmin@HCS-testingVM:~$ netcat -v -w 3 cloud.horizon.vmware.com 80
Connection to cloud.horizon.vmware.com 80 port [tcp/http] succeeded!
testvmadmin@HCS-testingVM:~$ netcat -v -w 3 cloud.horizon.vmware.com 443
Connection to cloud.horizon.vmware.com 443 port [tcp/https] succeeded!
testvmadmin@HCS-testingVM:~$ netcat -v -w 3 packages.microsoft.com 11371
Connection to packages.microsoft.com 11371 port [tcp/hkp] succeeded!
```

When a port is properly open, the `netcat` command returns the `succeeded!` line for its test.

If the `netcat` commands return failures, check your Microsoft Azure network connections, your Network Security Groups in your subscription, and any firewalls you might have in place. Ensure your network configuration meets the DNS, ports, and protocol requirements that the pod needs for deployment, as described in DNS Requirements for a Horizon Cloud Pod in Microsoft Azure.

**Results**

If the above tests succeed, you will be able to successfully deploy your pod.

**Note** If you will be configuring optional features for use with your pod, such as Radius two-factor authentication or True SSO, additional ports might be needed for those purposes. You can use the above outbound port testing techniques to verify such ports are properly open.
What to do next

When you have completed the test, you should delete the test VM and all of its related artifacts, such as its VM disk, IP address, NIC, from your Microsoft Azure environment. Ideally you would have created a resource group for the test VM and can simply delete that resource group to delete all of the VM's artifacts. Follow the steps in Delete the Test VM After You Complete the Tests.

**Important** If you do not delete all of the test VM's artifacts from your Microsoft Azure environment and you connected the VM to one of the pod's subnets, if you later try to delete the pod from your Horizon Cloud environment using the Delete action on the pod, the system might not be able to fully delete the pod due to those remaining connected artifacts. By default, when you use the Delete action to delete a pod, Horizon Cloud deletes those resource groups and subnets that it created for the pod. Microsoft Azure will prevent deletion of subnets that are still in use. If your test VM's artifacts are connected to the pod's subnets, then those subnets cannot be deleted and the pod deletion will be incomplete. To prevent this situation, ensure all of the test VM's artifacts are deleted after you have successfully deployed your pod.

Delete the Test VM After You Complete the Tests

When you have finished the tests to check your Microsoft Azure network configuration and no longer need the test VM, you should delete it and all of its related artifacts from your Microsoft Azure environment.

**Important** If you do not delete all of the test VM's artifacts from your Microsoft Azure environment and you connected the VM to one of the pod's subnets, if you later try to delete the pod from your Horizon Cloud environment using the Delete action on the pod, the system might not be able to fully delete the pod due to those remaining connected artifacts. By default, when you use the Delete action to delete a pod, Horizon Cloud deletes those resource groups and subnets that it created for the pod. Microsoft Azure will prevent deletion of subnets that are still in use. If your test VM's artifacts are connected to the pod's subnets, then those subnets cannot be deleted and the pod deletion will be incomplete. To prevent this situation, ensure all of the test VM's artifacts are deleted after you have successfully deployed your pod.

**Procedure**

1. Log in to the Microsoft Azure portal.
2 Use one of the following methods to delete the test VM, depending on how you deployed it.

- If you deployed the test VM into its own resource group and you are not using that group for any other purpose, you can delete the entire resource group.

**Caution** To avoid inadvertently deleting other items, make sure that the resource group contains only your test VM and its associated objects such as its disk and network adapters before deleting the resource group.

a In the portal's left hand navigation, click **Resource groups** and search for the test VM's resource group.

![Resource groups](image)

b Click the resource group's name to see the items in that resource group.

![Resource group items](image)

c Click **Delete resource group**. In the confirmation message, type in the resource group's name and then click **Delete**.
If you need to delete the test VM without deleting an entire resource group, you can use the portal's search box to search for the test VM's name. The results of this search will list the VM and all of its associated objects (disk, network interfaces, public IP address, and so on). Then delete each object individually.
Now That Your Very First Pod Is Completely Deployed and Connected to Horizon Cloud

Congratulations on deploying your first ever Horizon Cloud pod!

The console’s Getting Started page indicates when you have successfully created a cloud-connected pod.

The following screenshot illustrates what the page looks like when your very first pod is one deployed into Microsoft Azure.

At this point, you must perform the steps to register Horizon Cloud with the Active Directory domain that you want to use with this pod. The Horizon Cloud Administration Guide provides those detailed steps. See the topic named Getting Started Using Your Horizon Cloud Environment and its subtopics.