Deployments and Onboarding to Horizon Cloud for Microsoft Azure and Horizon Pods

Updated for the service starting from August 9, 2022
VMware Horizon Cloud Service
You can find the most up-to-date technical documentation on the VMware website at:
https://docs.vmware.com/
Horizon Cloud Connector 2.0 and Later - Add a Worker Node to a Horizon Cloud Connector Cluster 182

6 Horizon Cloud on Microsoft Azure Deployments - Key Characteristics 185
Get Started with a Simplified, Proof-of-Concept Horizon Cloud Service on Microsoft Azure Pod Deployment 191
Horizon Cloud on Microsoft Azure - First Pod Deployment - High-Level Workflow 273
Preparing to Deploy a Horizon Cloud Pod on Microsoft Azure 277
Microsoft Azure Virtual Machine Requirements for a Horizon Cloud Pod in Microsoft Azure 282
Configure the Required Virtual Network in Microsoft Azure 292
In Advance of Pod Deployment, Create the Horizon Cloud Pod’s Required Subnets on your VNet in Microsoft Azure 297
When Using Existing Subnets for a Horizon Cloud Pod in Microsoft Azure 300
Configure the DNS Server Settings Needed by the VNet Topology You Will Use for Your Horizon Cloud Pods in Microsoft Azure 303
DNS Requirements for a Horizon Cloud Pod in Microsoft and Related Service Features 304
Horizon Cloud Pod - Ports and Protocols Requirements 318
Create a Horizon Cloud App Registration in the Pod’s Subscription 336
When Your Organization Prefers to Use a Custom Role for the Horizon Cloud App Registration 343
Subscription-Related Information for the Horizon Cloud Pod Deployment Wizard 360
Resource Providers That Horizon Cloud Requires to Be in Registered Status in the Microsoft Azure Subscription 361
Convert a Certificate File to the PEM Format Required for Pod Deployment 363
Using the Horizon Universal Console to Perform an Automated Deployment of a Pod into Microsoft Azure 365
Prerequisites for Running the Pod Deployment Wizard 368
Start the Pod Deployment Wizard for Deploying a Pod-Manager-Based Pod 374
Specify the Microsoft Azure Subscription Information for the New Horizon Cloud Pod 377
Specify Pod Configuration Information for the Horizon Cloud Pod You Are Deploying Into Microsoft Azure Using the Deployment Wizard 382
Specify the Horizon Cloud Pod’s Gateway Configuration 388
Validate and Proceed, and then Start the Pod Deployment Process 403
Troubleshooting If You Encounter Pod Deployment or First-Time Domain Bind Issues 408
Horizon Cloud Pod Deployment Troubleshooting - Create an SSH Key Pair 410
Create the Test Virtual Machine in Your Microsoft Azure Subscription 413
Use SSH to Connect to the Test VM 416
Run the Tests to Check Networking in Your Microsoft Azure Environment 418
Delete the Test VM After You Complete the Tests 422

7 Now That Your Very First Pod Is Completely Deployed and Connected to Horizon Cloud 424
8 Revision History — Changelog — Onboarding to Horizon Cloud for Microsoft Azure and Horizon Pods 425

9 Deprecated - Manifests 1493.1 or Earlier - Horizon Cloud Pod - Ports and Protocols 432
Welcome Email, Initial License Purchaser, and the New Cloud Plane Tenant

The following is written from the point of view of license purchases. Trial requests typically follow the same sequence.

1. An initial purchaser purchases one of the Horizon subscription licenses. This initial purchaser might be yourself or might be another person in your organization who handles purchases.

2. In the control plane, VMware sets up the new tenant account and associates it with the initial purchaser’s VMware Customer Connect™ account and the purchased license information. If the purchase is also associated with a VMware Cloud services org or a Workspace ONE environment, VMware associates the tenant account with that information.

3. In that new tenant account, VMware also specifies one of the regional Horizon Cloud control plane instances. The information in the license purchase or trial 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.

4. VMware sends the Welcome to the VMware Horizon Service email to the initial purchaser’s email address. This email address is the one associated with that initial purchase order or
trial request. For an example of this welcome email, see the following screenshot. Among other information, the email states the 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.

The email contains information and hypertext links to key destinations. Those URLs link to the tenant environment portal (named the Horizon Universal Console or console for short), the Horizon Cloud Connector software download location, and online documentation.

The Subscription License

A subscription type of Horizon license is the initial requirement, because that is the point at which VMware generates your tenant account and configures the tenant environment in the service's cloud plane.

The license entitles you to use the cloud-plane services as well as use the subscription licensing with your pods. The day-1 onboarding flow is the key that enables you to exploit your Horizon subscription licenses, onboard your existing Horizon pods to use cloud-plane services, deploy new Horizon Cloud pods on Microsoft Azure, and leverage all of the cloud-plane services that VMware Horizon® Cloud Service™ currently provides for cloud-connected pods.

- A Horizon Cloud pod runs on the VMware Horizon Cloud pod-manager technology, which runs natively in Microsoft Azure along with the pod's virtual desktops. You onboard a Horizon Cloud pod on Microsoft Azure by using the Horizon Universal Console to deploy that type of pod into your Microsoft Azure cloud subscription.
A Horizon pod runs on the Horizon Connection Server software. This pod type requires VMware SDDC infrastructure for the pod’s virtual desktops, whether the VMware SDDC is an on-premises vSphere infrastructure or a cloud-based VMware SDDC environment such as Google Cloud VMware Engine (GCVE). You onboard such pods to the cloud plane using the Horizon Cloud Connector.

**About Your New Cloud-Plane Tenant Account**

Understanding your cloud-plane tenant account is important even when your only use case is to use a subscription license with your existing Horizon pods, while you are not yet planning to use cloud-hosted services with your pods. The reason why understanding this tenant account is important is because the same tenant account is used 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 Universal 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 account with which the license and access is associated. 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.

**Relating the Cloud-Plane Tenant to the Initial Purchaser’s VMware Customer Connect Account or VMware Cloud Services Account**

An account must be used to obtain the license in the first place — either a purchase or a trial order. The initial account is registered with the newly created tenant account and environment, and is used for authentication to the Horizon Cloud tenant environment's portals. When the tenant account is created, the **Welcome to the VMware Horizon Service** email is sent to the initial purchaser.

**Note** If you were not automatically made an administrator for your organization’s Horizon deployments when the tenant was provisioned, you can either request the original license acquirer to **add you as an administrator using the Horizon Universal Console** or you can file a non-technical support request in Customer Connect Support using the steps in **KB article 2006985**. In the support request, your account’s domain must match the domain in the initial purchase or trial order.
If you are the initial purchaser or trial requester and you are an administrator for your organization's Horizon deployments, then when you have received the email, you should log in to the tenant environment portal and add the additional administrators from your organization that you want involved in onboarding pods and managing your tenant. Adding additional admins to your tenant at the outset helps avoid access delays for your organization as a whole. Access to the tenant is needed to onboard pods, as well as to perform related workflows, such as reconfiguring the Horizon Cloud Connector. One admin can add additional admins for tenant access using the steps in Add Administrators to Log in to Your Horizon Cloud Tenant Environment.

The following screenshot illustrates a sample of the welcome email and calls out where the account is referenced. Notice that the email has helpful links guiding to downloads for the Horizon Cloud Connector, to open the administrative console, and so on.
Tip  The name My VMware was the former name of VMware Customer Connect. Both names are used interchangeably in emails, documentation, and the consoles for Horizon Service.
Accessing the New Cloud-Plane Tenant

As soon as the initial purchaser receives the welcome email from VMware, the account stated in that email can log in to the newly created cloud-plane tenant, even when the tenant has zero cloud-connected pods. If the initial purchase or trial order was done by a person who is not or will not be an administrator for your organization's Horizon deployments, then the people who will be administrators must request support to authorize them to log in to the new tenant environment. To make that request, file a non-technical support request in Customer Connect Support using the steps in KB article 2006985 to request your addition to your organization's existing tenant record. To be added this way, the domain in your account must match the domain in the initial purchase or trial order.

Additional ways in which a person in your organization might have access to the tenant is if they belong to an org in VMware Cloud service within which access to the service is granted, or if they belong to a Workspace ONE environment that is integrated with Horizon Cloud.

A brand new tenant environment, provides access to a single Horizon Universal Console 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 the tenant account is first created.

**Tip**  After logging in, you must click on the General Setup bar to see the key actions that are listed below.

Prior to onboarding your first pod, you can take these key actions from the rows in this screen.

**Capacity section**
If your tenant and login meets the license and role requirements, the **View perpetual keys** link is available to you on the console’s Getting Started page.

Clicking that link displays a UI screen in which you can view, copy, and generate perpetual keys for the foundational VMware products that are associated with your tenant.

In *Obtaining License Information Using the Horizon Universal Console*, you can read about the requirements for viewing, copying, and generating the keys for VMware foundational products that might be associated with your tenant.

**VMware SDDC row**

Click **Add** to learn how to download the Horizon Cloud Connector to connect a Horizon pod to your cloud tenant. This type of pod is based on the Connection Server software must already be deployed before you can connect it to your cloud-plane tenant. After clicking **Add** from this row, please ensure you follow the **Download** links that are displayed in the screen. Those steps are described in *High-Level Workflow When You are Onboarding a Horizon Pod*. Those steps will provide the smoothest experience at this time as they have links to information about prerequisites, DNS and ports requirements, and the precise sequence of steps to follow.

**Microsoft Azure row**

Start the automated wizard that will automate the deployment of a Horizon Cloud pod into your subscription in Microsoft Azure. These pods are based on the VMware Horizon Cloud pod-manager technology.

**General Setup section - My VMware Accounts**

Add the first set of administrators to whom you want to give the ability to log in to the Horizon Cloud Connector onboarding and configuration portal and the Horizon Universal Console (the portal to your tenant environment). The initial license purchaser’s account 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 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 purchaser’s account going
inactive for some reason — such as that person leaving your company or organization — it
is prudent to add the first set of administrators as soon as the Welcome to Horizon Service
e-mail is sent to the initial purchaser, even before the first pod is onboarded. If the initial
purchaser is not an administrator for your organization’s Horizon deployments, then at least
one person who will be an administrator must request support to authorize their VMware
Customer Connect account to the brand new service tenant account. After that person is
authorized, they can log in and add additional administrators using this My VMware Accounts
row. To make that request, file a non-technical support request using the steps in KB article
2006985.

---

**General Setup section - Cloud Monitoring Service (CMS)**

In the General Setup section, verify the Cloud Monitoring Service (CMS) settings you want. 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 deactivate 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.
Banner - Onboard to VMware Cloud Services Engagement Platform

When your Horizon Cloud tenant record is configured with the option to onboard to VMware Cloud services engagement platform and the tenant is not yet associated with an org in VMware Cloud services, a blue banner will appear at the top of the window that provides a way to activate that onboarding process. The following screenshot illustrates what you might see if your tenant record meets those conditions.

For information about that process, see Onboard Your Horizon Cloud Tenant to VMware Cloud Services. If you are accessing this portal from your Workspace ONE environment by clicking the Horizon Cloud tile, that blue banner will not appear. When your tenant record is not configured with the option to onboard to the platform, that blue banner will not appear.

Onboarding Requirements Checklists

When your initial pod onboarding is a Horizon pod deployment, to use subscription licensing or other cloud-based services

Review the items in the Chapter 4 VMware Horizon Pods with Horizon Cloud Control Plane - Requirements Checklist - Appropriately Updated Starting from the v2204 Service Release. That page describes the various prerequisite elements needed to successfully connect a Horizon pod to the cloud plane using the Horizon Cloud Connector.

When your initial pod onboarding is a Horizon Cloud on Microsoft Azure deployment

Review the items in the Chapter 3 VMware Horizon Cloud Service on Microsoft Azure Requirements Checklist For New Pod Deployments Starting From the v2207 Service Release. That page describes the various prerequisite elements needed before beginning the Horizon Cloud pod deployment wizard.

Revision History for these Onboarding Topics

This documentation set of topics is updated with each release of the product or when necessary. For the set of significant revisions made to date, see Chapter 8 Revision History — Changelog — Onboarding to Horizon Cloud for Microsoft Azure and Horizon Pods.

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 and Amazon Web Services EC2 (AWS EC2)
- Microsoft Azure and its Marketplace
- Azure VMware Solution (AVS)
- Google Cloud Platform (GCP) and Google Cloud VMware Engine (GCVE)

**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.


**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 VMware Customer Connect 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, you will see the following phrases. These phrases have the indicated meanings as follows.

**Horizon pod**

A pod that is constructed using the Horizon Connection Server software and related software components. The Horizon Connection Server components are running in an infrastructure that VMware supports for use with such pods. A Horizon pod typically requires a VMware SDDC (software-defined data center). Some examples of VMware SDDCs are an on-premises...
vSphere environment, VMware Cloud on AWS, Google Cloud VMware Engine (GCVE), or Azure VMware Solution (AVS), to name a few.

**Horizon Cloud pod, Horizon Cloud pod on Microsoft Azure**

A pod that is constructed by running the Horizon Cloud pod deployment wizard which automates deployment into your Microsoft Azure subscription. This type of pod is based on VMware Horizon Cloud pod-manager technology, and is supported to run in Microsoft Azure solely.

**Note** A Horizon pod on Microsoft Azure is separate and distinct entity from a Horizon pod on Azure VMware Solution (AVS). Those two are based on completely different technology — one is based on Horizon Connection Server technology and one is based on the Horizon Cloud pod-manager technology.

**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 receives the end-user's client request to make a connection with a virtual desktop VM or farm VM. The connection broker then routes the request appropriately and negotiates a connected session between the agent running in one of the VMs and that end-user client. The negotiation considers what types of pod-provisioned resources the end user is entitled to make connections with.

One of the Horizon Control Plane services is the Universal Broker service. Universal Broker is a multi-tenant, cloud-based service that enables the brokering of resources that span multiple pods and makes brokering decisions based on the geographic sites of users and pods.

A Horizon pod’s Connection Server or a Horizon Cloud pod’s pod manager VM is the component that facilitates the routing of the end-user client to a resource in the pod that meets the client’s connection request.

**About the Screenshots**

The screenshots typically:

- Show only a portion of the overall user interface screen, and not necessarily the full user interface. The depicted portion typically corresponds to the documentation text that describes that portion.
- 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, to see the image at its full-size, try clicking the screenshot.

**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.

**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](http://www.vmware.com/support/pubs).
Things to Know About Horizon Cloud and the Horizon Control-Plane Services

Use the following information and the linked-to articles when preparing to use Horizon Cloud or Horizon control-plane services, while onboarding your pods, and during day-to-day use. Refer back to this information throughout your journey using the control-plane services.

Set-Up Prerequisites, Software Downloads, User Settings Persistence, Product Documentation, and Additional Helpful Resources

Set-up prerequisites

For Microsoft Azure deployments, review the set-up prerequisites before you start deploying.

- VMware Horizon Cloud Service on Microsoft Azure Requirements Checklist
- Preparing to Deploy a Horizon Cloud Pod on Microsoft Azure

For onboarding your Horizon pods for cloud-plane services including the license service, review the descriptions of the type of deployment architecture your pod is using and the onboarding guidance. Refer to the information in the following locations:

- Chapter 5 Onboarding a Horizon Pod to Use Subscription Licenses or Control Plane Services with That Pod - for onboarding Horizon pods to the cloud plane.
- Horizon Reference Architecture pages at VMware Digital Workspace Tech Zone. Start at techzone.vmware.com/search#browser and select Horizon > Reference Architecture. Then select the deployment platform you are planning to use.
- Installation and Release Notes documentation, according to the pod’s Connection Server software version:
  - Version 7.13 - available at Horizon 7 Documentation.
  - VMware Horizon 8 versions - available at Horizon Documentation.

Software downloads

Review the software downloads you might want for your environment from VMware Customer Connect. Even though these downloads might be optional to have before you get started with your particular deployment, depending on your use case scenario, you might want to review them prior to deploying. See the VMware Horizon Cloud Service download page, locate the
most recent service release date, and navigate to its downloads link. Within that same page, you will see the Horizon Cloud Connector row and can click on its Go to Downloads to obtain the latest versions of both the Horizon Cloud Connector and VMware Universal Broker Plugin installer. For a table of which version of VMware Universal Broker Plugin Installer goes with which Horizon Connection Server, see the table in the topic Horizon Pods - Install the Universal Broker Plugin on the Connection Server.

User settings persistence

For all Microsoft Azure deployments, you can provide persistence of user profiles using VMware Dynamic Environment Manager™ with folder redirection. You can download the Dynamic Environment Manager software that is supported for use with this release from the VMware Horizon Cloud Service download page and navigate the downloads link for this specific release.

Product documentation and additional helpful resources

To access all of the product documentation for the various deployment models, see the VMware Horizon Cloud Service documentation landing page. Visit the community site for helpful tips and to ask any questions. Technical papers are also available in the Resources section of the Horizon Cloud product page.

Useful Day-0 Facts to Know

Before doing any deployment type

- When your Horizon Cloud environment is not integrated with your Workspace ONE environment, login authentication into the cloud-based console relies on authenticating account credentials with the VMware Cloud services platform. If that service is unable to complete the necessary authentication requests, logging into the console fails. If you encounter issues logging in to the console’s first login screen, check the VMware Workspace ONE Status page at https://status.workspace.com to see the latest system status. On that page, you can also subscribe to receive updates.

- When deploying a pod using the console’s pod deployer wizard and when connecting a Horizon pod using Horizon Cloud Connector, specific DNS names must be reachable and specific ports and protocols must be allowed. See DNS, Ports, and Protocols Requirements When Using Horizon Cloud Connector and a Horizon Pod, DNS Requirements for a Horizon Cloud Pod in Microsoft and Related Service Features, and Horizon Cloud Pod - Ports and Protocols Requirements for the connectivity requirements.

- Each of the pods paired with the Horizon Cloud control plane and associated with the same customer account must have line of sight to the Active Directory domains connected to those pods and have one-way or two-way trust configured along with that line of sight.
For example, when you have three pods where one pod is in Microsoft Azure, one pod is on-premises, and one pod in VMware Cloud on AWS, each of those pods must have line of sight and one-way or two-way trust configured to the same set of Active Directory domains.

Before doing Microsoft Azure deployments

- Subscriptions and Number of Pods: Be mindful about the number of pods you deploy into a single Microsoft Azure subscription, especially if you plan to have each pod running at a large scale. Even though multiple pods can be deployed into a single Microsoft Azure subscription, whether all into one region or spread across multiple regions, Microsoft Azure imposes certain limits within a single subscription. Because of those Microsoft Azure limits, deployment of many pods into a single subscription increases the likelihood of hitting those limits. Numerous variables, and combinations of those variables, are involved in reaching those limits, such as the number of pods, the number of farms and assignments within each pod, the number of servers within each pod, the number of desktops within each assignment, and so on. If you plan to have pods running at a large scale, consider adopting the approach of having multiple subscriptions with those multiple subscriptions under one Microsoft Azure account. Microsoft Azure customers can, and often prefer, this approach because it provides some benefits for ongoing management of the subscriptions. Using this approach, you would deploy a single pod per subscription, roll up those subscriptions in a single "primary account", and avoid the chances of hitting the Microsoft Azure limits that are imposed on a single subscription.

- Outbound Internet access is required on the Microsoft Azure Virtual Network (VNet) used by the deployment’s pod manager VMs. Proxy-based authentication is supported for a Horizon Cloud on Microsoft Azure deployment. You must provide your proxy details in the pod deployment wizard. For pod deployment, specific DNS names must be reachable and specific ports and protocols must be allowed. See DNS Requirements for a Horizon Cloud Pod in Microsoft and Related Service Features for the connectivity requirements.

- Subnet sizing: Expanding the size of the pod’s subnets after the pod is deployed is not currently supported. As a result, for production environments, you should use subnet sizes that are large enough to accommodate the following requirements:
  - Management subnet: When deploying a pod, as of March 2019, the pod’s management subnet is required to have a minimum of CIDR /27, where in previous releases a lower minimum CIDR of /28 was allowed. This change was made to reduce the occurrence of issues that can happen during pod updates due to lack of available IP addresses in the subnet. A CIDR of /27 provides for 32 IP addresses.
  - VM subnet - primary: Use a CIDR in a range that is large enough to accommodate attaching the VMs for your anticipated VDI desktops, the RDS images, and every VM in the pod's RDS farms. The pod manager VMs and the Unified Access Gateway VMs
also need some IP addresses from this subnet (12 addresses total to accommodate
the blue-green update of an HA-enabled pod with both types of gateways). Generally
speaking, the range of /24 to /21 would provide for typical use cases. Note: At times,
this VM subnet is referred to as the desktop subnet or the tenant subnet.

- Starting with service release July 2020 and pod manifest 2298.0, a new feature
  provides for using additional tenant subnets for your VDI desktops and RDS farm VMs.
  Those additional subnets can be in the same VNet as the pod or in peered VNets.
  For a pod at manifest 2298.0 or later, you can edit the pod’s configuration to include
  those additional subnets. Then you can specify use of those additional tenant subnets
  in the definitions of your farms and VDI desktop assignments instead of them using
  the primary VM subnet. Use of these secondary subnets for your farm VMs and VDI
  desktop VMs provides for simplified administration, because you can specify which
  farms and VDI desktop assignments are on which tenant subnet and VNet.

- To leverage the feature to deploy the external gateway into its own VNet, the VNets
  must be peered. As a result, you must create the subnets manually in advance of running
  the deployment wizard. For the external gateway’s VNet, its management subnet and
  back-end subnet must each adhere to the same minimum CIDR /27.

Before connecting pods using Horizon Cloud Connector

- New deployments should download and use the latest version of Horizon Cloud Connector
  that is available at VMware Customer Connect and which is compatible with the pod's
  Horizon Connection Server software version. Using the latest version ensures you have the
  most recent fixes and improvements. For the compatibility matrix between Horizon Cloud
  Connector and Horizon Connection Server, visit VMware Product Interoperability Matrix
  and check interoperability between the two solution names listed as VMware Horizon
  Cloud Connector and VMware Horizon.

- Outbound Internet access is required for the Horizon Cloud Connector to communicate
  with the service's cloud plane, especially to receive the licensing details. Specific DNS
  names must be reachable and specific ports and protocols must be allowed. See DNS,
  Ports, and Protocols Requirements When Using Horizon Cloud Connector and a Horizon
  Pod for the connectivity requirements.

- Before connecting a second Horizon pod to Horizon Cloud, you should log in to
  the Horizon Cloud administrative console and complete the Active Directory domain
  registration process after connecting your first Horizon pod using the Horizon Cloud
  Connector’s onboarding process. If you pair multiple Horizon pods with Horizon Cloud
  before completing that Active Directory domain registration, unexpected results might
  occur when you eventually log in to the console to attempt the domain registration
  process.

- Due to a known issue, when you are using an on-premises Active Directory domain to
  service a pod in VMware Cloud on AWS, slow access times might occur due to network
  latency or network congestion between that on-premises Active Directory domain and the
pod in VMware Cloud on AWS which results in calls to the domain timing out. Symptoms of this latency typically include the Active Directory login screen failing to complete the login before timing out. If you experience such symptoms, configuring a writable domain controller in each in-cloud software-defined data center (SDDC) might help.

About Some of the Service's Operational Emails

Starting with the April 2021 service release, for every customer record that is associated with a designated VMware customer account representative, some of the system's operational emails will by default include the email addresses of those designated VMware customer account representatives in the BCC field. The purpose for including the VMware customer account representatives in these emails’ BCC field is to ensure better onboarding and business continuity.

The operational emails that will include the VMware customer account representatives in the BCC field are:

- At the initial creation of your customer record for the service, a customer name and email address is specified as Owner in the newly created customer record. A welcome email is sent to the email address associated with that Owner customer name.
- When any updates occur related to that Owner — such an updated email address — a notification email is sent.
- When any updates occur related to the customer record’s Horizon licenses, a notification email is sent.

This chapter includes the following topics:

- **Horizon Cloud — Available Environments, Operating System Support, Tight Integration Within the VMware Ecosystem, and Compatibility Information**
- **Support for Windows 11 Guest Operating System - Considerations, Known Limitations, and Known Issues**
- **For Current Customers with Existing Cloud-Connected Pods — About Horizon Cloud Service Releases**
- **VMware Horizon Cloud Service on Microsoft Azure Service Limits**
- **Active Directory Domain Configurations**
- **Horizon Cloud — Known Limitations**
- **Horizon Cloud — Known Issues**
Horizon Cloud — Available Environments, Operating System Support, Tight Integration Within the VMware Ecosystem, and Compatibility Information

This documentation topic provides information about the environments and operating systems available for use with Horizon Cloud. This topic also describes the tight integration that Horizon Cloud has with the VMware ecosystem, and a convenient pointer to the VMware Interoperability Matrix.

Microsoft Azure Cloud Environments

For Microsoft Azure deployments, the service is currently available in the following Microsoft Azure cloud environments.

- Microsoft Azure (Commercial)
- Microsoft Azure in China
- Microsoft Azure Germany (Public)

Supported Microsoft Windows Operating Systems

In the service's Microsoft Azure deployments, the following Microsoft Windows operating system editions and versions in the Azure Marketplace are the ones supported for use in this release, regardless of whether you use the automated or manual method of deploying an image in this release.

- For the supported non-Windows 10 or 11 operating systems, see VMware Knowledge Base Article 78170.
- For the supported Windows 10 or Windows 11 operating systems, see VMware Knowledge Base Article 70965.

The service also provides Tech Preview support for Windows 7 virtual desktops with Extended Security Updates. With virtual desktops based on this operating system, you can use the Horizon Windows client, RDP 8.x protocol, and the Horizon remote experience features USB redirection and Help Desk. This Tech Preview support for Windows 7 does not include use with GPU-capable VMs. Use of Windows 7 and GPU-capable VMs in Horizon Cloud is unsupported.

The service's support for Windows 11 currently has some known considerations, limitations, and issues. For those details, see Support for Windows 11 Guest Operating System - Considerations, Known Limitations, and Known Issues.
Tight Integration Within the VMware Ecosystem

You can use Horizon Cloud with the following products available from the broader VMware ecosystem.

**VMware Carbon Black**

For information on integrations between VMware Carbon Black Cloud and using the VMware Carbon Black sensor with farms and VDI desktops provisioned by your Horizon Cloud pods, see [VMware Carbon Black Interoperability with Horizon Cloud Service on Microsoft Azure (KB 81253)](https://kb.vmware.com/kb/81253).

**VMware Workspace ONE® Hub Services and VMware Workspace ONE® Access™ Cloud**

With this integration, the desktop and remote application assignments that are brokered using VMware Horizon® Service Universal Broker™ will synch automatically to the VMware Workspace ONE® Intelligent Hub catalog and be made available through the Hub for your end users to view and launch. In the current release, this integration supports end-user access using these clients: browser-based Hub catalog, Workspace ONE Intelligent Hub for Windows, and Workspace ONE Intelligent Hub for macOS. The minimum version of the Windows and macOS desktop apps required for this support is 21.05.

**VMware Workspace ONE® UEM**

You can use Workspace ONE UEM to manage the Microsoft Windows 10 dedicated VDI desktops provisioned by your Horizon Cloud pods. See [Using Workspace ONE UEM and Your Dedicated VDI Desktops](https).

**VMware Workspace ONE® Assist™ for Horizon®**

Workspace ONE Assist for Horizon enables Horizon Cloud administrators to launch remote support sessions directly from the Help Desk Tool located in the Horizon Universal Console. With this product, administrators can assist employees with their virtual desktops, leveraging remote view and control capabilities. For all details, see the [Workspace ONE Assist for Horizon and Horizon Cloud document](https), located within the Workspace ONE UEM product documentation.

*Note* Integrating with Workspace ONE Assist for Horizon involves additional DNS, ports, and protocol requirements for the outbound communication from the involved VDI desktops. See the [Workspace ONE Assist for Horizon and Horizon Cloud document](https) for those details.

**VMware NSX-T™ Data Center**

The compatibility between Horizon Cloud on Microsoft Azure versions and NSX-T Data Center versions is now available in the VMware Product Interoperability Matrix, at [https://interopmatrix.vmware.com](https://interopmatrix.vmware.com) for the most recent versions of Horizon Cloud Service on Microsoft Azure. The Horizon Cloud documentation reflects the latest configuration steps, according to the versions shown as compatible in the [VMware Product Interoperability Matrix](https).
If you have pods still running manifests from Horizon Cloud Service on Microsoft Azure versions prior to version 2010 and you have already configured use of NSX-T Data Center 2.4 or 2.5 with such pods, the configuration will continue to work. However, you should update such pods when the Horizon Cloud system indicates in the console that those pods should be updated.

**VMware NSX® Advanced Load Balancer™**

Horizon Cloud supports using NSX Advanced Load Balancer with the gateway configurations of your Horizon Cloud pods in Microsoft Azure. NSX Advanced Load Balancer provides the Avi Vantage load-balancing features from Avi Networks, now part of VMware. The reference design, sequence of steps, and applicable configuration information is provided in the Avi Networks article titled *Load Balancing UAGs in Horizon Cloud on Azure Deployments*.

**Note** Some aspects of this integration with your Horizon Cloud pods might require assistance from the VMware Horizon Cloud Service team to complete the overall configuration. To obtain the latest information before starting this integration, file a service request (SR) to the Horizon Cloud Service team as described in *How to file a Support Request in Customer Connect (VMware KB 2006985)*.

**VMware Horizon® Client™ product line**

To see the specific versions of the Horizon Client product line that are compatible with the desktops and remote applications brokered by the pods in your cloud-connected pod fleet, see *VMware Product Interoperability Matrix*. Depending on the pod type, select either Horizon Cloud Service on Microsoft Azure or VMware Horizon, and then VMware Horizon Client in the drop-down menus.

Generally speaking, the various flavors of Horizon Client and the VMware Horizon HTML Access client do not necessarily support the same set of features or protocols for every use case. As of this writing, some of these variations are:

- The VMware Horizon HTML Access client does not support certain features when used in mobile browsers. Also, even though the Horizon Client supports copying and pasting text between a client's local system and a VM out of the box, for the web client, you must configure this feature before your end users can use it.

- The VMware Horizon Client for Mac does not provide a menu option to use the RDP protocol.

- The Horizon Clients do not support using the RDP protocol with RDSH applications.

To download a matrix of supported features between Horizon Client and Horizon Cloud Service on Microsoft Azure, see the *VMware KB article Horizon Client Feature Matrix for*...
Horizon Cloud on Azure (80386). Please note that keeping that matrix up to date is on a best-effort basis at this time.

**Note**
- Use of zero clients with Horizon Cloud pods in Microsoft Azure is not supported.
- Use of zero clients with Universal Broker is not supported.

When using an operating-system-specific Horizon Client to access remote resources brokered by Universal Broker, the following client versions are supported. You must provide your end users with the connection FQDN for Universal Broker to use in the client. For instructions on how to configure that connection FQDN, see Configure the Universal Broker Settings.

- Horizon Client 5.4 or later for their operating system.
- Windows users can run Horizon Client for Windows 5.3 or later.

End users can also connect to the Universal Broker service through a web browser using Horizon HTML Access. Note that in the following cases when using the web-based client, end users will see the standard browser ‘unsafe’ message when they start a brokered desktop:

- When the pod’s Unified Access Gateway setup’s load balancer is configured with a SSL certificate that does not have a common name that precisely matches that load balancer’s name or is not signed by a well-known Certificate Authority (CA).
- When the pod has an external Unified Access Gateway, but no internal Unified Access Gateway setup, and the end user’s access is over the internal network.
- When the pod has no Unified Access Gateway setups at all.

(For information about the relationship between a certificate's common name and the hostname of where the certificate is installed, see https://support.dnsimple.com/articles/what-is-common-name/.)

**VMware Horizon® Pods**

A Horizon pod is the pod type that is built on the Horizon Connection Server software. You connect such pods to Horizon Cloud using Horizon Cloud Connector. New deployments of Horizon Cloud Connector are supported using versions \( N, N-1, N-2 \), where \( N \) is the latest generally available Horizon Cloud Connector version. The version number of the latest generally available version of Horizon Cloud Connector appears at the top of the Horizon Cloud Service Release Notes document.
For the matrix of Horizon pod software that corresponds with those \( N, N-1, N-2 \) versions of Horizon Cloud Connector, see the VMware Product Interoperability Matrix. To get the benefits of advanced features that are only available for cloud-connected Horizon pods, the deployment must use the most recent versions of the Horizon Cloud Connector and Horizon pod software. You can download the latest Horizon Cloud Connector appliance and VMware Universal Broker plugin installer by navigating to the Horizon Cloud Connector section located within the VMware Horizon Cloud Service download page.

**Note**  As of January 2021, Horizon Cloud Connector versions earlier than 1.6.x will be unable to connect to the cloud control plane. In late 2020, all existing tenants with deployments based on earlier versions were notified to update their connectors.

### Compatibility with other VMware Products

For the most recent information about compatibility between this product and other VMware products, see the VMware Product Interoperability Matrix.

### About Use of IPv6

Horizon Cloud does not support use of IPv6.

### About TLS 1.2 and Horizon Cloud Service on Microsoft Azure

New deployments of Horizon Cloud pods on Microsoft Azure have TLS 1.2 set as the minimum TLS version for both the pod's storage accounts and the Azure PostgreSQL Service that is deployed as part of the service offering. These storage accounts and the Azure PostgreSQL Service are used only with the service components such as the pod manager instances. Therefore, the pod's use of TLS 1.2 should have no impact on your customer-managed artifacts in your Microsoft Azure subscription.

### Browser Experience

The cloud-based administrative console is compatible with recent versions of Google Chrome, Mozilla Firefox, and Microsoft Edge. Use of the console in Microsoft Internet Explorer 11 is deprecated and will give a sub-optimal experience. The console is not supported for use in Apple Safari, although you can try using the console in Apple Safari. If you attempt to access the console using a non-modern browser such as Microsoft Internet Explorer 11, the console displays an information message to use an up-to-date browser. For the best user experience, use the most recent versions of Google Chrome, Mozilla Firefox, and Microsoft Edge.

### Support for Windows 11 Guest Operating System - Considerations, Known Limitations, and Known Issues

The following considerations, limitations and issues have been identified for the Windows 11 Guest operating system.
Considerations

- Pod must be running the manifest from v2204 release or later.
- Golden images must be running Horizon Agent Installer v22.1.0 or later.
- Support matrix for Gen 1/Gen 2 and Windows 10/Windows 11.

<table>
<thead>
<tr>
<th>Azure VM Model</th>
<th>Windows 10</th>
<th>Windows 11</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gen 1 VM</td>
<td>Supported</td>
<td>Not Supported</td>
</tr>
<tr>
<td>Gen 2 VM</td>
<td>Not Supported</td>
<td>Supported</td>
</tr>
</tbody>
</table>

Limitations

- The console-based App Volumes app capture workflow is currently unsupported. To use App Volumes on Azure with Windows 11 VDI desktops, run the console-based capture workflow using a Windows 10 golden image to capture the apps. Then assign those apps to users to use with a Windows 11-based desktops.
- Manual import of a Windows 11 image requires importing from Azure Marketplace as the direct source. Importing from any other sources such as Shared Image Gallery (SIG), Azure Managed Images, Azure VM snapshot, and the like are currently unsupported.
- vTPM is currently unsupported.
- Use of Windows 11 with VMs running AMD drivers is currently unsupported.

Known Issues

- When time zone redirection is enabled using GPO, flickering desktop and explorer process crashing occurs. See KB 88086 for details.
  
  Avoid the known issue by not enabling time zone sync GPO for Windows 11 multi-session VMs.

For Current Customers with Existing Cloud-Connected Pods — About Horizon Cloud Service Releases

Use the information in this page in tandem with the What's New in the Release Notes. Along with the items listed in the Release Notes document, this page is for those of you who already have existing cloud-connected pods onboarded in your environment prior to the most recent service refresh date, or you have previous experience with Horizon Cloud features and workflows. This page describes the significance the newly debuted features and changes might mean to you and those pods. Only significant changes to the features and workflows are described. Minor changes, such as new layouts and color schemes in the administrative console that do not significantly change the workflows are not detailed here.

The most recent service refresh date is listed in the service's Release Notes document. To see the up-to-date information about the various workflows that you perform in your Horizon Cloud tenant environment, read the documentation topics found in the individual guides linked from the Horizon Cloud Service documentation page.
The following sections go back to September 2019. Similar information for earlier releases is not available for publication.

**Important** For information that is key to understanding terms used in this page, see the information in the [Release Notes document](#) prior to reading the information in this page. The Release Notes document has key relevant pieces of information such as the latest pod manifest number for pods that deploy into Microsoft Azure. Also keep in mind that the key facts described in the following section are true in every Horizon Cloud release.

### Six Key Facts About Every Horizon Cloud Release

- All new cloud-plane-based features that are agnostic to pod manifest version level or Horizon Cloud Connector version level or Horizon pod version level or control-plane regional differences are automatically provided to both existing customers and new customers. As an example, a new feature that does not depend on new APIs for API calls between the cloud plane to the pods or related to Horizon Cloud Connector will be visible and exploitable by existing customers unless otherwise noted below or noted in the product guides. To illustrate this point, one such feature is the console's enhanced feedback submission. Even though this feedback icon debuted in July 9, 2020, the icon and flow is available for use by tenant environments that existed prior to July 9, 2020 even prior to updating the pods or Horizon Cloud Connector to their latest versions.

- The VMware Horizon Cloud Service team debuts new cloud-plane-based features that are agnostic to pod manifest version level or Horizon Cloud Connector version level or Horizon pod version level on an ongoing, regular basis and on any week of the calendar year. The [Release Notes document](#) contains the debut dates of those features.

- Starting on the day on which that manifest version debuts in the cloud control plane, the pod deployer for deploying pods into Microsoft Azure always deploys pods at the latest pod manifest version.

- Already deployed pods that exist in your service tenant prior to the day on which a new pod manifest version debuts in the cloud plane will continue running at their existing manifest version until they are updated to the manifest that is brand new for the release. The following facts apply:

  - New service features which have zero dependencies on APIs that require the latest manifest level will be available for those existing pods.
  
  - New service features which rely on APIs at the latest manifest level will not be available for those existing pods until those pods are updated.

  - Some new service features might depend on the cloud plane region in which your tenant account is located. Such features are noted in the documentation where applicable. Your control-plane region is stated in the Welcome to Horizon Service email sent when the customer account is created, as described in [Deployments and Onboarding to Horizon Cloud for Microsoft Azure and Horizon Pods](#).
The console's Import VM from Marketplace wizard uses the Horizon Agents Installer (HAI) that is built into the pod manifest. As a result, pods deployed at the latest manifest version will have the latest HAI built into them, and running the Import VM wizard and selecting a pod at the latest level will install the agents from that latest HAI. For pods that are not yet updated to the latest manifest level, the Import VM from Marketplace wizard uses the HAI version that was available when their respective pod manifests were built.

Horizon pods are onboarded to Horizon Cloud for two primary use cases: to activate use of a subscription license with those pods and to enable use of cloud-hosted services that Horizon Cloud provides for Horizon pods. Each pod is onboarded using the Horizon Cloud Connector. These use cases debuted starting with Horizon 7 version 7.6 environments and Horizon Cloud Connector 1.0 for activating subscription licenses on Horizon pods. Then with each new version of Horizon Connection Server combined with a new version of Horizon Cloud Connector, additional cloud-hosted services become available for cloud-connected Horizon pods running the latest Horizon Connection Server version paired with the latest Horizon Cloud Connector version. New deployments of Horizon Cloud Connector are supported using versions N, N-1, N-2, where N is the latest Horizon Cloud Connector version. We recommend that deployments that are still using versions of Horizon Cloud Connector that are earlier than the N-2 version to update to the latest versions to take advantage of new features as well as security and resiliency fixes. For the latest N version of Horizon Cloud Connector, see the upper portion of the Release Notes document. Also see the VMware Product Interoperability Matrices tool for the matrix of currently supported versions of Horizon pod software with Horizon Cloud Connector. If you are running a combination of Horizon Connection Server and Horizon Cloud Connector version that no longer matches that matrix, please update as soon as possible to a supported combination.

**August 2022 - v2207**

Use the following information when you are an existing customer with cloud-connected pods dating from before August 2022 and you want to understand any effects on your experience from the features described in the Release Notes for August 2022 and the v2207 release.

To request enablement of an enabled-by-request feature, file a support request as described in VMware KB article 2006985.

**New versions of the following key binaries have debuted in v2207:**

- A new pod manifest version for Horizon Cloud on Microsoft Azure deployments and the Horizon Agents Installer (HAI).

**New items related to service-deployed pods in Microsoft Azure**

- Administrators can now configure the cipher suites that will be accepted when clients connect to the Unified Access Gateway machines. For pods with gateway configurations that existed in your tenant prior to this release, use the pod details page to review the cipher suites configured on those gateway configurations. You can use the Edit Pod workflow to change the configured cipher suites.
- During upgrade maintenance, Unified Access Gateway session counts will be used to optimize timing to reduce end-user session interruptions.

- The transient pod deployment engine, also known as the jump box, is removed from the architecture for new Horizon Cloud on Microsoft Azure deployments and updates. Capacity for a transient jump box would be required only if you open a support request and VMware Support determines the way to service that request is to deploy a support-related jump box VM, under their supervision.

- To avoid effects from Microsoft Azure Cloud's upcoming retirement of some VM models, better accommodate importing GPU-capable VMs, and standardize the VM models for single-pod and multi-pod image imports, starting in the v2207 release, the VM models that the service's automated Import VM from Marketplace wizard uses by default have changed.

  The wizard now uses the following models as indicated for both single-pod images and multi-pod images. We recommend that you check the VM family quotas in your pods' Azure subscriptions to ensure that you have available quota for the images you plan to create using the wizard.

  **Import VM from Marketplace wizard creates:**

  - Non-GPU, non-Windows 11, a Standard_DS2_v2 VM
  - Non-GPU using Windows 11, a Standard_D4s_v3 VM
  - GPU-capable, a Standard_NV12s_v3 VM

**New items related to Horizon Image Management Service**

- Horizon customers with deployments in VMware Cloud on AWS using all-in-SDDC architecture can now replicate images using the Multi-Pod Images capabilities. Along with using the standard IMS features for markers and pools with the deployment in VMware Cloud on AWS, with this support, you can use IMS to replicate images from Horizon on-premises deployments to the Horizon in VMware Cloud on AWS all-in-SDDC deployments and from the VMware Cloud on AWS deployments to the on-premises deployments.

  For the requirements for the support of IMS with all-in-SDDC Horizon deployments in VMware Cloud on AWS, see the relevant information in Horizon Image Management Service System Requirements.

**Additional items of note**

- Because every subscription includes a Workspace ONE Access tenant which can be added using the Horizon Universal Console, the capability of new Workspace ONE Access tenant creation from within the Horizon Universal Console. This change supports VMware business operations.
May 2022 - v2204

Use the following information when you are an existing customer with cloud-connected pods dating from before May 2022 and you want to understand any effects on your experience from the features described in the Release Notes for May 2022.

To request enablement of an enabled-by-request feature, file a support request as described in VMware KB article 2006985.

New versions of the following key binaries have debuted in v2204:


New items related to service-deployed pods in Microsoft Azure

- Windows 11 is now a supported guest operating system for use with Horizon Cloud on Microsoft Azure deployments. Refer to this page for known considerations, limitations, and issues when using Windows 11 with the service, Support for Windows 11 Guest Operating System - Considerations, Known Limitations, and Known Issues. Use of this feature requires the pod running this release's manifest or later.

Some additional points related to this Windows 11 support:

- If you use the manual import method to import a VM with Windows 11, you must ensure that the Horizon Agents Installer (HAI) version 22.1 or later is used to install the agents. The Windows 11 support requires the minimum agent version provided by HAI v22.1.

- Use of the console's App Volumes Capture workflow using a Windows 11 golden image is not currently supported. As a work around, you can capture the packages using a Windows 10 golden image and then assign those packages to your end users to use with their assigned Windows 11 single-session or multi-session desktop.

- New Horizon Cloud on Microsoft Azure pods will always be deployed with high availability enabled.

- Multi-pod images will now leverage configured proxy settings for any image operation originating from the customer network to the internet (such as when using the automated Import Virtual Machine from Marketplace wizard).

- Support for manually importing VMs from the Azure Marketplace that use AMD GPU and graphics drivers and using those imported VMs for golden images. This support requires use of the Azure VM model Standard_NV4as_v4 and the pod running this release's manifest or later.

- In existing dedicated assignments, you can now adjust a provisioned VM's workload CPU, memory, or disk based on individual user needs. When you do this, the pool type will change to Mixed type. Use of this feature is supported for pods running this release's manifest or later.
The recommended Horizon Agents Installer version is now displayed on the console's Capacity screen. Regular notifications will be generated as reminders to keep agents up to date. This feature is provided for pods running this release's manifest or later.

The service created Microsoft Azure VMs that enter a Stopped state due to a guest OS shutdown will now automatically be transitioned to a Deallocated state to prevent continued billing. Use of this feature is supported for pods running this release's manifest or later.

Enhancements in the area of App Volumes for Horizon Cloud pods. To use this feature, the pod must be running this release's manifest level or later.

Applications can now be delivered on-demand when a user clicks to launch the application from the desktop and Start menu, and the resulting behavior is the same as if the application was already installed natively on the Windows machine. The setting is available for newly created packages.

To mitigate the effects from a Microsoft Azure issue, starting in the v2204 service release, the VM models that the service's automated Import VM from Marketplace wizard uses by default for non-GPU Windows 10 OS images have changed. For details of the reported issue, refer to VMware KB88343.

The wizard also uses specific models for the Windows 11 support that is newly added in this v2204 release.

The wizard now uses the following models. We recommend that you check the VM family quotas in your pods' Azure subscriptions to ensure that you have available quota for the images you plan to create using the wizard.

**Single-pod image - Import VM from Marketplace wizard creates:**

- Non-GPU Windows 10 OS or a Windows 10 Enterprise multi-session OS single-pod image, a Standard_DS2_v2 VM
- Non-GPU Windows Server OS VM single-pod image, a Standard_D2_v3 VM
- GPU-capable Windows-10 OS or a Windows 10 Enterprise multi-session OS or a Windows Server OS single-pod image, a Standard_NV6 VM
- Non-GPU Windows 11 OS or a Windows 11 Enterprise multi-session OS single-pod image, a Standard_D4s_v3 VM
- GPU-capable Windows 11 OS or a Windows 11 Enterprise multi-session OS single-pod image, a Standard_NC6s_v3 VM
- Non-GPU Windows 7 OS single-pod image, a Standard_DS2_v2 VM (GPU unsupported on Windows 7)

**Multi-pod image - Import VM from Marketplace wizard creates:**

- Non-GPU Windows-10 OS or a Windows 10 Enterprise multi-session OS or a Windows Server OS multi-pod image, a Standard_DS2_v2 VM
- GPU-capable Windows-10 OS or a Windows 10 Enterprise multi-session OS or a Windows Server OS multi-pod image, a Standard_NV6 VM
- Non-GPU Windows 11 OS or a Windows 11 Enterprise multi-session OS multi-pod image, a Standard_D4s_v3 VM
- GPU-capable Windows 11 OS or a Windows 11 Enterprise multi-session OS multi-pod image, Standard_NC6s_v3 VM
- Non-GPU Windows 7 OS multi-pod image, a Standard_DS2_v2 VM (GPU unsupported on Windows 7)

Additional items of note

- In the IMS support for Horizon on-premises pods:
  - An option is provided to select the datastores and networks in the target pod's vCenter Server for the image copy.
  - Use of IMS with a multi-clustered vCenter Server is now supported.
  - A Republish option to retry publishing when a publish fails is now provided for this pod type.

March 2022 - v2203

Use the following information when you are an existing customer with cloud-connected pods dating from before March 2022 and you want to understand any effects on your experience from the features described in the Release Notes for March 2022.

To request enablement of an enabled-by-request feature, file a support request as described in VMware KB article 2006985.

**Note** The generally available versions of Horizon Agents Installer, Horizon Cloud Connector, and Universal Broker Plugin Installer are unchanged in this v2203 release.

New versions of the following key binaries have debuted in v2203:

A new pod manifest version for Horizon Cloud pods deployed into Microsoft Azure by the service's deployment wizard. In addition to providing the backend support that the pod managers need for some of the debuted features, this manifest includes platform code improvements for reliability.

New items related to service-deployed pods in Microsoft Azure

- A new Edit button on a VDI desktop assignment's Summary tab provides convenience for editing the assignment’s details.
- A farm's Summary tab displays additional useful information, such as the farm's creation date and time, its Azure resource group name, and its farm ID.
- The pod deployer no longer requires creation of an Azure Key Vault during pod deployment. Please note that the Add Pod wizard will continue to validate that the
subscription has the Microsoft.KeyVault resource provider in Registered state, to support use of the farms’ and VDI desktop assignments’ disk encryption feature. That feature requires creation of a key vault in the pod manager’s resource group at the time you select use of that disk encryption feature.

**Important** Existing customers should not delete the key vaults that exist in their Horizon Cloud on Microsoft Azure deployments unless under direction by VMware Support. Manually deleting the key vaults will put the deployments into an unsupported state.

- RSA SecurID is an option for a two-factor authentication configuration on the pod’s gateway configurations. To use this feature with an already deployed gateway configuration, the pod must be running manifest 3139 or later. This feature is targeted for enablement in the Add Pod and Edit Pod wizards by mid-March 2022. The option will be visible in those wizards at that time.

**Additional items of note**

- Use of the Horizon Infrastructure Monitoring features with Horizon pods is no longer supported. As a result, all references to using those features with Horizon pods is removed from documentation. (Horizon pods are the ones that run Connection Server.)

**February 2022 - v2201**

Use the following information when you are an existing customer with cloud-connected pods dating from before February 2022 and you want to understand any effects on your experience from the features described in the Release Notes for February 2022.

**Note** The generally available versions of Horizon Agents Installer, Horizon Cloud Connector, and Universal Broker Plugin Installer are unchanged in this v2201 release.

**New versions of the following key binaries have debuted in v2201:**

A new pod manifest version for Horizon Cloud pods deployed into Microsoft Azure by the service’s deployment wizard. In addition to providing the backend support that the pod managers need for some of the debuted features, this manifest includes platform code improvements for performance and reliability.

**New items related to service-deployed pods in Microsoft Azure**

- The gateway configurations now support configuring a syslog server to receive the logs from the Unified Access Gateway instances. This feature is supported for pods running manifest 2298 or later.

- LDAPS can now be selected as the protocol when registering your Active Directory environment with your tenant. This feature is available when your tenant is explicitly enabled for it and all pods are running this release’s manifest level. To request enablement, you must file a support request as described in VMware KB article 2006985.
During the time that an upgrade is scheduled for a pod, the pod's details page in the console displays a banner which indicates that after the upgrade of the pod and its Unified Access Gateway instances, you might need to update the IP addresses that are configured in your RADIUS server as allowed client connections from the NICs on the Unified Access Gateway instances. When a pod has a gateway configuration, Unified Access Gateway instances are also updated when that pod is updated to a new manifest version. With this feature, during the time period that the upgrade is scheduled, the pod's details page in the console will alert you that when the gateway configuration has configured RADIUS settings, you might have to take additional action to ensure that your RADIUS server configuration is updated to allow client connections from the IP addresses that the gateway NICs will be using post-upgrade.

**New items related to the Image Management Service (IMS)**

- You can now select which pods you want the multi-pod images to be copied to. Previously, the images were copied to all pods by default.

**New items related to Universal Broker**

- In the help desk tool, the breakdown of logon segments is displayed in the session data for users connecting through Universal Broker.
- When using Universal Broker and Dynamic Environment Manager together, Dynamic Environment Manager can now distinguish between internal and external users, for the purposes of applying smart policies.
- For Horizon pods, the VDI multi-cloud assignment workflow now supports pools that are running Windows Server 2019.

**November 2021 - v2111**

Use the following information when you are an existing customer with cloud-connected pods dating from before October 2021 and you want to understand any effects on your experience from the features that debuted in the service with this month’s release.

New versions of the following key binaries have debuted in November 2021: a new pod manifest version for service-deployed pods, new versions of Horizon Cloud Connector, Universal Broker Plugin Installer, and the Horizon Agents Installer (HAI).

**New items related to service-deployed pods in Microsoft Azure**

- You can now edit the VM Model type for an existing VDI desktop assignment. The pod must be running this release’s manifest level.
- Multi-pod image management now supports multi-session operating systems. You can now create farms from such multi-pod images. The pod must be running manifest 2915.x or later.
In tenants enabled with single-pod broker, you can now move an individual VM between VDI desktop assignments that are provisioned by the same Horizon Cloud pod. To have use of this feature, your tenant must be explicitly enabled for it and the pod must be running this release's manifest level. You can request enablement of this feature in your tenant by filing a support request as described in VMware KB article 2006985.

In the area of auto-agent updates, you can now remediate incomplete or failed agent updates in cases when the agent is stopped and not running in the VM. The pod must be running this release's manifest level.

Enhancements in the area of App Volumes for Horizon Cloud pods. The pod must be running this release's manifest level.

- App packages will now be automatically detached as the last assigned user of that app logs off a Windows 10 Enterprise multi-session system. Previously, a shutdown of the VM was required to detach the packages.
- App Volumes now supports VMware Dynamic Environment Manager with Windows 10 Enterprise multi-session operating systems.
- App Volumes batch files can be used to configure the App Volumes workflow. This feature is covered in the App Volumes Administration Guide in the page titled Batch Scripts for App Volumes workflow.
- The RunAsUser.exe utility is now included with App Volumes Agent, allowing you to run the executable from the App Volumes advanced configuration batch files. Typically, the code from these batch files executes in the system scope and you can use this utility to run code in the context of the current logged in user. This feature is covered in the App Volumes Administration Guide in the page titled Use RunAsUser for App Volumes Batch Scripts.
- Improvements in the Print Spooler restart behavior in a Windows 10 Enterprise multi-session operating system.

New items related to Universal Broker

Starting with this release, for greenfield deployments of Horizon Cloud on Microsoft Azure Universal Broker is the enabled broker by default.

Additional items of note for current customers

- You can now specify email addresses to receive alerts and notifications from your tenant and those addresses are not required to be associated with a tenant administrator role. Prior to this feature, only tenant administrators could receive alerts and notifications by email.
- When your tenant’s Horizon Universal License includes licensing for VMware SDDC components such as VMware vCenter, vSAN, and vSphere, you can retrieve those keys
using the Horizon Universal Console. In the console, you can select which versions of the keys you want to retrieve and can view those keys at any time. To use the console to generate the key, you must have the Super Administrator role in the tenant and be a Customer Connect user on the EA that is associated with this tenant.

- When Workspace ONE redirection is enabled at the tenant level, end users are appropriately redirected to Workspace ONE Hub even when the end-user clients are directly connecting to the FQDN on the pod's Unified Access Gateway configuration. Previously, such end-user connections were directly obtaining a desktop.

October 2021 - v2110

Use the following information when you are an existing customer with cloud-connected pods dating from before October 2021 and you want to understand any effects on your experience from the features that debuted in the service with this month's release.

Note  The generally available versions of Horizon Agents Installer, Horizon Cloud Connector, and Universal Broker Plugin Installer are unchanged in this v2110 release.

Also, the pod manifest version for Horizon Cloud pods deployed into Microsoft Azure by the service’s deployment wizard remains as 3000.x. If critical fixes are required, patches such as 3000.1, 3000.2, and so on will be made available.

New items related to service-deployed pods in Microsoft Azure

- Horizon Universal Console now supports narrow scope permissions to desktop assignments and farms for the built-in predefined roles. To have use of this feature, your tenant must be explicitly enabled for it and all of the Horizon Cloud pods must be running manifest 2915.x or later. You can request enablement of this feature in your tenant by filing a support request as described in VMware KB article 2006985.

- In the New Pod workflow and Edit Pod workflow, administrators can now choose 8443 as the TCP port for Blast Extreme on the pod’s Unified Access Gateway instances. Prior to this feature, the Blast Extreme TCP port was set to 443 by default and there was no wizard option to change it. Port 8443 is strongly preferred because it has better performance for the client-to-gateway traffic and uses less resources on the Unified Access Gateway instances. The chances of CPU congestion in the instances causing traffic delays from the client are reduced when you use 8443 for the Blast Extreme TCP port. You can use the Edit Pod workflow to change the Blast Extreme TCP port for existing gateway configurations.

- For App Volumes applications with multi-session desktops, application package detachment now happens after the last user assigned that package logs off. Shutting down the underlying farm VM is no longer required to detach volumes. Previously, the application package detachment happened at shut down of the VM.

New items related to Universal Broker

- Universal Broker and multi-cloud assignments now support brokering of desktops and apps for pods using Google Cloud VMware Engine (GCVE).
Universal Broker now supports the ability to restrict the launching of virtual desktops, published desktops, and published applications to specific clients and versions and provide warning messages to clients.

Universal Broker now supports the ability for end users to connect to their VDI desktops and published desktops using the RDP protocol from within those Horizon Clients that provide the RDP protocol as an option for use with Windows-based desktops.

New items related to Horizon Cloud Connector and cloud-connected Horizon pods

With the debut of the Horizon Cloud Connector version 2.0 native binary for Google Cloud Platform, all of the cloud-plane services that the cloud plane provides for Horizon pods deployed with a federated architecture are now supported for such pods. Such pods use Google Cloud Platform for the management components and Google Cloud VMware Engine (GCVE) for the desktop components. Use Horizon Cloud Connector version 2.0 to obtain this full support. Please note that currently the Image Management Service (IMS) is only supported with on-premises Horizon pods and not for cloud-based Horizon pod deployments.

With the debut of the Horizon Cloud Connector version 2.0 native binary for Amazon EC2, all of the cloud-plane services that the cloud plane provides for Horizon pods deployed with a federated architecture are now supported for such pods. Such pods use Amazon EC2 for the management components and VMware Cloud on AWS for the desktop components. Use Horizon Cloud Connector version 2.0 to obtain this full support. Please note that currently the Image Management Service (IMS) is only supported with on-premises Horizon pods and not for cloud-based Horizon pod deployments.

Horizon Cloud Connector version 2.0 OVA is now supported for use with Horizon on VMware Cloud on Dell EMC. These pods are all-in-SDDC deployments. Except for the Cloud Monitoring Service (CMS) and Image Management Service (IMS), those cloud-plane services that the cloud plane provides for Horizon pods in all-in-SDDC deployments are supported for this deployment type.

September 2021 - v2109

Use the following information when you are an existing customer with cloud-connected pods dating from before September 2021 and you want to understand any effects on your experience from the features described in the Release Notes for September 2021.

Note The generally available versions of Horizon Agents Installer, Horizon Cloud Connector, and Universal Broker Plugin Installer are unchanged in this v2109 release.

New versions of the following key binaries have debuted in v2109:

A new pod manifest version for Horizon Cloud pods deployed into Microsoft Azure by the service's deployment wizard. In addition to providing the backend support that the pod
managers need for some of the debuted features, this manifest includes platform code improvements for performance and reliability.

**New items related to service-deployed pods in Microsoft Azure**

- Horizon Agent Update now supports targeting agent updates on individual desktops within the assignment. Use of this feature requires a pod running at this release's manifest level.

- The pod deployment wizard and Edit Pod wizard no longer mandate having a Unified Access Gateway configuration unless the tenant's Universal Broker settings include two-factor authentication. Previously, the wizards enforced the use of at least one Unified Access Gateway on the pod.

- When your tenant is configured to use single-pod brokering, connecting to a VDI desktop or a farm-based session desktop using the RDP protocol is now supported. This feature is supported for pods running manifest 3000.x or later.

- The system requirements for transitioning a single-pod broker tenant to Universal Broker no longer include having at least one internal or external Unified Access Gateway configuration on every pod except in the following scenarios.
  - When you select to enable two-factor authentication in the Universal Broker settings in the Schedule Transition wizard. When two-factor authentication is enabled for Universal Broker in the wizard, the transition requires that every pod in the tenant’s fleet have an external Unified Access Gateway.
  - When your tenant’s fleet also includes Horizon pods, and the console's Broker page indicates their Universal Broker configuration already has two-factor authentication enabled. In this case also, the transition requires that every pod in the tenant’s fleet have an external Unified Access Gateway.
  - For App Volumes applications with multi-session desktops, application package detachment now happens after the last user assigned that package logs off. Shutting down the underlying farm VM is no longer required to detach volumes. Previously, the application package detachment happened at shut down of the VM.

**New items related to Horizon Cloud Connector and cloud-connected Horizon pods**

- Now available are documented steps for using the high availability features of vSphere with the Horizon Cloud Connector appliance. Because this ability relies on the vSphere high availability features, it is specific to the on-premises and all-in-SDDC pod architectures. In those deployment architectures, the Horizon Cloud Connector is deployed into a vSphere infrastructure.

- Use of Cloud Monitoring Service and pods deployed with the federated deployment design that uses Google Cloud Platform and Google Cloud VMware Engine (GCVE) is now supported.
The Change to Managed State wizard no longer mandates having a Unified Access Gateway configuration unless the tenant's Universal Broker settings include two-factor authentication. Previously, the wizard enforced the use of at least one Unified Access Gateway on the pod.

**New items related to Universal Broker**

For end users on your internal network, Universal Broker now supports direct connections between their clients and the virtual desktops and remote applications (VDI and RDSH). With this support, an internal Unified Access Gateway is no longer required to launch virtual desktops and remote applications for these internal connections. To support these internal clients' launch of the virtual desktops and remote applications, you must use the console's `Broker > Network Ranges` to specify ranges of egress NAT addresses so that Universal Broker can recognize the specified ranges as originating from your internal network.

**August 2021 - v2108**

Use the following information when you are an existing customer with cloud-connected pods dating from before August 2021 and you want to understand any effects on your experience from the features described in the Release Notes for August 2021.

**Note**  The generally available versions of Horizon Agents Installer, Horizon Cloud Connector, and Universal Broker Plugin Installer are unchanged in this v2108 release.

**New versions of the following key binaries have debuted in v2108:**

A new pod manifest version for Horizon Cloud pods deployed into Microsoft Azure by the service's deployment wizard. This manifest includes platform code improvements for performance and reliability.

**New items related to service-deployed pods in Microsoft Azure**

- Two additional operations are added to the set of operations that the service principal needs to use in your subscription, as described in When Your Organization Prefers to Use a Custom Role for the Horizon Cloud App Registration. These two additional operations are to support an upcoming feature by which the service can reduce the time it takes for a new pod deployment and for a pod upgrade by using pre-configured images in the Microsoft Azure Marketplace. When your service principal uses a custom role, that role will need the ability to perform these two additional operations.

- When creating an App Volumes assignment, the wizard no longer includes the OS Family selection menu. Because packages captured in a VM running one type of Windows operating system are compatible with desktops running the types of Windows operating systems that are available for use with these pods, specifying the OS family for the assignment was determined to be no longer necessary. Therefore, the menu could be removed from the wizard with no loss of functionality. This change is also applicable to
App Volumes assignments that were created prior to this release. When editing such existing App Volumes assignments, even though the **OS Family** was selected when that assignment was created, because the **OS Family** choice is no longer required by the system, that field is no longer present in the edit window.

- Additional improvements in the area of reporting on failures that occur during the automated agent update process. When the pod is running this release's new manifest version, the downloadable CSV report now also provides the names of the desktop VMs in which the agent update process was successful or skipped, in addition to the ones in which the process has failed.

**Additional items of note for current customers**

Integration with a new VMware product, VMware Workspace ONE Assist for Horizon, enables Horizon Cloud administrators to launch remote support sessions directly from the Help Desk Tool from the Horizon Universal Console. Administrators can assist end users with tasks and issues involving their virtual desktop sessions. VMware Workspace ONE Assist for Horizon is part of the VMware Workspace ONE UEM product line. For all of the requirements for using the features provided by this new product, see the topic 'Requirements for Horizon Cloud with Workspace ONE Assist' in the **Workspace ONE for Horizon and Horizon Cloud** document.

The Contact Information section of the Horizon Universal Console General Settings page is removed because its fields are not needed for any workflows or use cases or system operations.

**July 2021 - v2106**

Use the following information when you are an existing customer with cloud-connected pods dating from before July 2021 and you want to understand any effects on your experience from the features described in the **Release Notes** for July 2021.

New versions of the following key binaries have debuted in July 2021: a new pod manifest version for service-deployed pods, new versions of Horizon Cloud Connector, Universal Broker Plugin Installer, and the Horizon Agents Installer (HAI).

**New items related to Horizon Cloud Connector and its use with Horizon pods**

- Version 2.0 of Horizon Cloud Connector adds service-level fault tolerance for the license service. Documentation about this feature can be found within the Administration Guide topic **Horizon Cloud Connector 2.0 - Clusters and Service-Level Fault Tolerance** and its subtopics.
- Version 2.0 of Horizon Cloud Connector adds support for monitoring the appliance using SNMP. Administrators can use this standards-based monitoring function to proactively
monitor and receive alerts on critical connector-related services, such as licensing, upgrade, and connector life cycle, even if they are not logged into the Horizon Universal Console. Documentation about this feature can be found within the Administration Guide topic Horizon Cloud Connector 2.0 - Monitor the Appliance Using SNMP.

New items related to service-deployed pods in Microsoft Azure

- Multi-pod image management provides an easy method for publishing and replicating single-session VDI images to two or more Horizon Cloud pods in Microsoft Azure and updating multiple VDI multi-cloud assignments with a single operation. The features of multi-pod image management with Horizon Cloud pods is only available to tenants configured to use VDI multi-cloud assignments and for Horizon Cloud pods running manifest 2632.x and later. Documentation about this feature can be found in Managing Horizon Images From the Cloud and its subtopics.

- Enhancements in the area of automated agent updates:
  - Prior to a scheduled agent update, the system will now by default automatically restart the VMs selected for the agent update process. This automated restart helps to ensure the VM and the software and services running inside its operating system are in a known state prior to running the agent update operations.
  - Improved reporting on failures that occur during the automated agent update process. You can download a CSV report of the names of the desktop VMs in which the agent update process has failed, even when the agent update process is still underway on the assignment. The CSV report is available for download when the pod is running this release's new manifest version.

May 2021 - v2105

Use the following information when you are an existing customer with cloud-connected pods dating from before May 2021 and you want to understand any effects on your experience from the May 2021 release.

**Note** The generally available versions of Horizon Agents Installer, Horizon Cloud Connector, and Universal Broker Plugin Installer are unchanged in this v2105 release.

**New versions of the following key binaries have debuted in v2105:**

A new pod manifest version for Horizon Cloud pods deployed into Microsoft Azure by the service's deployment wizard. This manifest includes platform code improvements for performance and reliability.

**New items related to Horizon Cloud Connector and its use with Horizon pods**

Universal Broker now supports published desktops and applications for Horizon pods on VMware SDDC. For this support, the pods must be running VMware Horizon 8 software version 2103 or later and Universal Broker Plugin Installer version 21.03 or later.
April 2021 - v2104

Use the following information when you are an existing customer with cloud-connected pods dating from before April 2021 and you want to understand any effects on your experience from the April 2021 release.

**Note**  The versions of Horizon Cloud Connector and Universal Broker Plugin Installer are unchanged in this v2104 release. You can continue to use the March 2021 versions of those components with your tenant environment.

**New versions of the following key binaries have debuted in v2104:**

- A new pod manifest version for Horizon Cloud pods deployed into Microsoft Azure by the service's deployment wizard. This manifest includes platform code improvements for performance and reliability.
- A new HAI version includes a fix to resolve an intermittent issue related to the Cloud Monitoring Service receiving data from the Horizon agent. (problem report 2742816).

**New items related to service-deployed pods in Microsoft Azure**

- Support for moving App Volumes packages between App Volumes applications using the console. Includes support for moving newly imported packages to the appropriate applications.

**Additional items of note for current customers**

- Some features debut as Limited Availability features. Such features are enabled on a tenant-by-tenant basis, usually on a per-request basis.

March 2021 - v2103

Use the following information when you are an existing customer with cloud-connected pods dating from before March 2021 and you want to understand any effects on your experience from the new features.

New versions of the following key binaries have debuted in March 2021: a new pod manifest version for service-deployed pods, new versions of Horizon Cloud Connector, Universal Broker Plugin Installer, and the Horizon Agents Installer (HAI).

**New items related to Horizon Cloud Connector and its use with Horizon pods**

- Version 1.10 of Horizon Cloud Connector brings fixes and security updates.

**New items related to service-deployed pods in Microsoft Azure**

- Universal Broker and multi-cloud assignments are now available for existing deployments of Horizon Cloud pods on Microsoft Azure.
- The console's pod detail's page has a new Maintenance feature, used to specify a preferred maintenance window. Use this feature to inform the system of your preferred time and weekday for pod maintenance activity to occur.
Starting with this release's pod manifest, App Volumes for Horizon Cloud pods on Microsoft Azure now supports Microsoft Windows 10 Enterprise multi-session, allowing multiple users to each login into individual sessions with their own app assignments. Previously, the use of Microsoft Windows 10 Enterprise multi-session with App Volumes for Horizon Cloud pods was in Tech Preview. Please note that the feature is supported for deployments starting with this release's pod manifest and starting with this release's App Volumes Agent version.

For customers using App Volumes for Horizon Cloud pods on Microsoft Azure and also using the App Volumes Command-Line Capture Program to capture applications for importing into your Horizon Cloud applications inventory, a new option has debuted with App Volumes 4 version 2103 that provides for packaging applications without needing the App Volumes Manager console, as well as other tools to work with VHD formats and with the MSIX app attach formatted packages. For information, see the App Volumes 4 version 2103 Release Notes What's New.

Additional items of note for current customers

Some cloud-plane-based features debuted in the weeks prior to the v2103 release.

Universal Broker and multi-cloud assignments now support Horizon pods on Azure VMware Solutions (AVS), enabling unified brokering of multi-cloud assignments across hybrid and multi-cloud deployments, supporting both Horizon pods and Horizon Cloud pods on Microsoft Azure.

The console is enhanced to reflect its new name: Horizon Universal Console.

Some features have debuted as Limited Availability features. Such features are enabled on a tenant-by-tenant basis, usually on a per-request basis.

In the administration console for Horizon pods (known as Horizon Console), the Cloud Brokered option is not operational for this release. This option appears in the RDS desktop pool settings and application pool settings for cloud-managed Horizon pods running VMware Horizon 8 version 2103 (8.2). Enabling the Cloud Brokered option has no effect for this Horizon Cloud release.

January 2021 - v2101

Use the following information when you are an existing customer with cloud-connected pods dating from before January 2021 and you want to understand any effects on your experience from the new features.
New versions of the following key binaries have debuted in January 2021: a new pod manifest version for service-deployed pods, new versions of Horizon Cloud Connector, Universal Broker Plugin Installer, and the Horizon Agents Installer (HAI).

New items related to Horizon Cloud Connector and its use with Horizon pods

- With Horizon Cloud Connector 1.9, the automatic update feature now has a simpler configuration for the network details. You only need to provide an unassigned static IP address for the new appliance for upgrade.
- Horizon Cloud Connector 1.9 enables more secure access methods when troubleshooting the Horizon Cloud Connector appliance. SSH access for the appliance's root user is deactivated and a new custom user (ccadmin) is now available for SSH access, including support for using an SSH public key instead of password credentials.

New items related to service-deployed pods in Microsoft Azure

The pods now support the ability to rollback the dedicated desktop to a previous usable state in the event that the agent update fails and a configurable failure threshold that provides a fail-fast mechanism that will stop the update process and skip any remaining desktops. For this feature to be used with a previously existing pod, that pod must first be updated to manifest 2632.0 or later.

Additional items of note for current customers

Some cloud-plane-based features debuted in the weeks prior to the v2101 release.

For pod manifests of 2474.0 and later, the set of Active Directory permissions for the domain join accounts is reduced to provide more flexibility for tenants. However, because the system will use the same domain join accounts in domain-join-related operations with all of the pods in your tenant's fleet, when your fleet includes pods of manifests prior to 2474.0, you must ensure your domain join accounts have the prior set of permissions that those pods require. When all of your Horizon pods in Microsoft Azure are updated to pod manifest 2474.0 or later, then you can adopt the newer reduced set of Active Directory permissions for your domain join accounts. See Service Accounts That Horizon Cloud Requires for Its Operations and the updated section about the permissions for the domain join account.

The console is enhanced to reflect some new terminology. Previously, the console displayed labels such as On-Premises and VMware Cloud on AWS when referring to cloud-connected Horizon pods that are installed in on-premises vSphere installations or in a VMware Cloud on AWS SDDC (software-defined data center). These are the pods that are based on the Horizon Connection Server software. With the availability of Horizon pod deployments on additional cloud-hosted VMware SDDCs such as Horizon pod on Azure VMware Solution (AVS) and Horizon pod on Google Cloud VMware Engine (GCVE), the console is enhanced to use the label VMware SDDC when referring to those members of your tenant's pod fleet.

Support is added for running the Enrollment Server on Microsoft Windows Server 2019. The Enrollment Server is used for the True SSO feature. Installation of Enrollment Server on Microsoft Windows Server 2008 is no longer supported.
October 2020 - v2010

Use the following information when you are an existing customer with cloud-connected pods dating from before October 2020 and you want to understand any effects on your experience from the new features.

New versions of the following key binaries have debuted in October 2020: a new pod manifest version for service-deployed pods, new versions of Horizon Cloud Connector, Universal Broker Plugin Installer, and the Horizon Agents Installer (HAI).

**New items related to Horizon Cloud Connector and its use with Horizon pods**

- Horizon Cloud Connector version 1.8 is released in both OVA and VHD forms.
- Horizon Cloud Connector 1.8 provides the ability to select a deployment profile to either enable with subscription license support only or with Horizon Cloud features. This selection is made during the deployment of the appliance.
- Horizon Cloud Connector now supports Horizon pods that are deployed on Azure VMware Solution (AVS). Currently, this support is for use of your subscription license with those deployments. The full set of cloud-hosted services are not yet provided for these deployment types.

**New items related to service-deployed pods in Microsoft Azure**

Horizon Cloud on Microsoft Azure pods now support the ability to specify custom Azure Resource Tags during a pod deployment or gateway deployment. The pod deployer applies the specified tags on the resource groups that the pod deployer creates. For a description of the resource groups that the pod deployer creates, see Resource Groups Created for a Pod Deployed in Microsoft Azure. This new feature is not dependent on the pod manifest version.

July 2020 - v3.1

Use the following information when you have cloud-connected pods dating from before July 2020 and you want to understand any effects on your experience from the new features.

**Specifically about existing cloud-connected Horizon pods**

- Debut of Horizon Cloud Connector 1.7.

**Specifically about existing pods in Microsoft Azure**

- For the following new features to be used with a previously existing pod, that pod must first be updated to manifest 2298.0 or later to take advantage of the feature.
  - Multiple tenant subnets for use with farms and VDI desktop assignments. This feature is not yet available for use with multi-cloud desktop assignments, which are used in a tenant configured with Universal Broker.
  - Use of the advanced session load balancing for RDSH farms.
Ability to cancel both desktop and farm expansion tasks that are in a queued or running state, with support for automatic desktop assignment and farm resizing. This feature is not yet available for use with multi-cloud desktop assignments, which are used in a tenant configured with Universal Broker.

To provide for improved end-user login times, the time it takes for a VM to get to agent-ready state for pod-provisioned desktop VMs that are powered off and need to power on to fulfill an end user’s request for a desktop has been reduced.

Use of the App Volumes features — the pods must be updated to this release’s manifest version and your customer account must be located in one of the following Horizon Cloud control-plane regions: USA-2 (PROD1_NORTHCENTRALUS2_CP1), Europe-2 (PROD1_NORTHEUROPE_CP1), or Australia-2 (PROD1 AUSTRALIAEAST_CP1). Your control-plane region is stated in your Welcome to Horizon Cloud Service email.

When deploying a gateway configuration on your pod, in addition to the Standard_A4_v2 VM size in previous releases, you now have the option to use the Standard_F8s_v2 VM size, which provides more vCPUs for each Unified Access Gateway instance. For existing pods, this new feature is available when editing the pod to add a new gateway configuration to that pod.

Additional items of note for current customers
An enhancement to submitting product feedback is now available in the console’s header bar, for all existing customers.

Pods that can have the high availability (HA) feature are now supported for Microsoft Azure Government (US Gov Virginia, US Gov Arizona, US Gov Texas). If you have an existing pod in Microsoft Azure Government for which you want this feature, please contact your VMware representative for that enablement.

March 2020 - v3
Use the following information when you have cloud-connected pods dating from before March 2020 and you want to understand any effects on your experience from the new features.

Specifically about existing cloud-connected Horizon pods
The debut of Horizon Cloud Connector 1.6.x provides a command-line diagnostic tool for you to check the health of required Horizon pod’s system components and services that are needed for Horizon Cloud Connector to successfully pair the pod with Horizon Cloud. Before you log in to the web-based configuration portal and run the pod configuration wizard, you can run this diagnostic tool to check for things that might prevent a successful outcome. If there are issues discovered, the tool will report the component name, details, and recommended remediation steps.

Specifically about existing pods in Microsoft Azure
For the following new features to be used with a previously existing pod, that pod must first be updated to manifest 1976.0 or later to take advantage of the feature, unless otherwise noted.

- To support advanced deployment configurations, when using a separate subscription for the external Unified Access Gateway configuration, you can choose to deploy the Unified Access Gateway resources into an existing customer-created resource group, instead of the default one created by the pod deployer. For an existing pod to take advantage of this new feature, the pod must first be updated to at least manifest version 1763 or later (the December 2019 manifest). Then you need to meet all the documented requirements for using a separate subscription, VNet, and custom resource group for the external gateway configuration, including peering that VNet with the pod’s VNet and creating the resource group in that subscription. Then you must delete the pod’s existing external Unified Access Gateway configuration, using the console’s workflow to delete that existing external gateway. When the deletion is completed successfully, then you can run the Edit Pod workflow to add back an external gateway using the new option to place the external Unified Access Gateway in your existing resource group.

- Support for administrators to specify that the names of the dedicated VDI desktop assignments are displayed in the end-user clients after a VDI desktop VM gets assigned to the end user, instead of displaying the desktop VM name. Previously, after an end user claimed a specific VDI desktop VM, their client displayed the name of the desktop VM by default and that behavior was not configurable. This option does not change what is displayed for those end-user connections that are going through Workspace ONE Access. Workspace ONE Access always displays the dedicated VDI desktop assignment name, and when the end user launches the desktop VM from Workspace ONE Access, the desktop name is displayed in their end-user client. Even though you will see this feature’s option in the General Settings page, a pod must be running manifest 1976.0 or later for it to be able to take advantage of this feature.

- Pod manifest 1976.0 or later supports administrators to put an individual farm VM into a maintenance mode, so that the administrator can perform maintenance actions on the VM. Before you can exploit this feature to set a per-VM maintenance mode, the pod must be running this release’s manifest version. Also, due to a known issue in the console, even though this feature’s options are displayed in the farm’s Servers tab in the console, those user-interface options will not set the mode until the agents in the farm’s VMs are running version 20.1.0 or later.

Additional items of note for current customers

Enhancements in the reports available in the console’s Reports page and Dashboard page. The data in these reports is provided by the Cloud Monitoring Service. Existing pods can take advantage of this feature.
December 2019 - v2.2

Use the following information when you have cloud-connected pods dating from before December 2019 and you want to understand any effects on your experience from the new features.

Specifically about existing cloud-connected Horizon pods

Starting in this release:

- Some things that are within your control can prevent a successful automatic update of the Cloud Connector, such as insufficient datastore space in your vCenter environment to accommodate the update. Starting in this release, if automated update is enabled for your Horizon Cloud tenant account, such items are identified in the console, so that you can address and clear those items.

- Automated updates of Horizon Cloud Connector are now supported for Horizon pods deployed in VMware Cloud on AWS.

- Enhancements to the Horizon Cloud Connector onboarding success screen include a health status display for the connector's components and an option for activating and deactivating SSH on the Horizon Cloud Connector appliance.

Specifically about existing pods in Microsoft Azure

For the following new features to be used with a previously existing pod, that pod must first be updated to manifest 1763.0 or later to take advantage of the feature, unless otherwise noted.

- To support advanced deployment configurations, the pod deployer provides options for:
  - Using a separate VNet for the external gateway configuration's Unified Access Gateway instances, separate from the pod's VNet and the core pod elements. The VNets must be peered.
  - Using a separate subscription for the external Unified Access Gateway configuration, separate from the subscription used for the core pod elements. Because a VNet is scoped to a subscription, the separate subscription deployment scenario is also the separate VNet scenario. The VNets must be peered.
  - For an existing pod to take advantage of this feature, the pod must first be updated to manifest 1763.0 or later. Then you need to meet all the documented requirements for using a separate VNet for the external gateway configuration, including peering that VNet with the pod's VNet. Then you must delete the pod's existing external Unified Access Gateway configuration, using the console's workflow to delete that existing external gateway. When the deletion is completed successfully, then you can run the Edit Pod workflow to add back an external gateway using the new options.

- Starting with this release's manifest, you can use SSD disk types for VDI desktop assignments and RDSH farms.
Starting with this release's manifest, you can customize the OS disk sizes for VDI desktop assignments and RDSH farms. At earlier pod manifests, their OS disk sizes were set to be the same as the published base image, which was 127 GB by default and could not be changed.

New in this release, in the Import VM from Marketplace wizard, you will see a toggle that provides the ability to omit joining the resulting VM to an Active Directory domain. Previously, this workflow joined the VM to the domain by default and you could not change that behavior. This new toggle is available for existing pods prior to this release's manifest version.

With the redesign of the Capacity page in this release, the Type view is removed. With the removal of the Capacity page's Type view, there are two changes to note about items that were previously accessed from that view: the action to view the pod's current usage of its subscription's Microsoft Azure limits has moved to the pod's details page and the Remove Subscription action that had been present in that view is removed completely.

Additional items of note for current customers

- Enhancements in the reports available in the Administration Console's Reports page. The data in these reports is provided by the Cloud Monitoring Service.

- Enhancements to the Horizon Cloud Administration Console's Capacity page. Instead of having to drill-down into a pod's details page to modify the pod's configurable details or to delete a pod from your tenant environment, you can now initiate the edit pod and remove pod workflows from the Capacity page itself. As a result of this redesign, the workflows for modifying location information that were previously done using the Capacity page's Location view are now options within the Edit Pod workflows. As an example, to specify a new location name, use the Edit action on a pod and you can specify a new location name as an option of that Edit Pod workflow. Please note that the previous Location view's workflow for removing saved Microsoft Azure subscription information when all of that subscription's associated pods have been deleted is no longer available.

- The product name formerly known as VMware Identity Manager is now named VMware Workspace ONE™ Access.

- The Horizon Agents Installer no longer installs a dormant DaaS agent. In the previous release, the HAI installed the DaaS agent's MSI into the guest operating system, but it was dormant and not used. In this release, the MSI is not installed at all.

September 2019 - v2.1

Use the following information when you have cloud-connected pods dating from before September 2019 and you want to understand any effects on your experience from the new features.

Specifically about existing cloud-connected Horizon pods
Starting in this release:

- Automatic update is now supported on Cloud Connector versions 1.3 and 1.4. It is recommended that customers on earlier versions of Cloud Connector update to the latest to take advantage of this feature.
- Cloud Monitoring Services (CMS) with session usage details is provided as part of Horizon Cloud Service.

Specifically about existing pods in Microsoft Azure

For the following new features to be used with a previously existing pod, that pod must first be updated to manifest 1600.0 or later to take advantage of the feature, unless otherwise noted.

- The pod architecture has changed in this release. All pods at the September 2019 release’s manifest version have a pod Microsoft Azure load balancer and a Microsoft Azure Database for PostgreSQL server instance (Gen 5 Memory optimized tier). This means that before you update your existing pods to this release’s manifest version, you must ensure that your existing networking configuration meets the DNS, ports, and protocols required to accommodate the pod Microsoft Azure load balancer and Microsoft Azure Database for PostgreSQL server instance. If you have firewalls or network security groups that block specific ports and protocols, compare your current networking configuration to the information in the following topics and update your networking configuration accordingly.
  - DNS Requirements for a Horizon Cloud Pod in Microsoft Azure
  - Ports and Protocols Requirements for a Horizon Cloud Pod at the September 2019 Release’s Manifest Level

- Enhanced alerting for pod update errors that require customer actions to resolve them is provided in this release. Some things that are completely within your control can prevent a successful pod update, such as not having enough cores in the pod’s associated subscription to create the jump box VM which orchestrates the pod update. Starting in this release, such items are identified in the console, so that you can address and clear those items.

- Starting in this release, you can revise the following gateway-related settings on an already deployed pod: add two-factor authentication settings to a gateway that does not have them, edit a gateway’s two-factor authentication settings, change the gateway’s session brokering timeout setting. In previous releases, you had to configure RADIUS two-factor authentication when the pod was first deployed, and could not change those settings afterwards. Also new in this release, you can delete gateways from an already deployed pod and you can deploy a new pod to have an external gateway without a public IP address on its Azure load balancer and instead have a private IP address on that load balancer.

- Support for defining Microsoft Azure resource tags when creating a new dedicated or floating VDI desktop assignment or a new farm for Horizon Cloud on Microsoft Azure.
High availability is now available. To support high availability for pods in Microsoft Azure, the pod architecture is updated to use the Microsoft Azure Database for PostgreSQL service (Updated Gen 5 Memory optimized tier), a Microsoft Azure load balancer, and an availability set. For a pod that is newly deployed in this release, you have the option to enable high availability for that pod at deployment time, or enable high availability later on. For pods that existed prior to this release, before you can enable those pods for high availability, you must first update them to the 1600.0 manifest or later and also update the agents in the pods’ images, farms, and VDI desktop assignments to this release level. When the pod update and agent updates are completed, then you can enable high availability on the pod by editing the pod from its pod details page in the Administration Console. This new feature brings additional requirements for enabling the Microsoft.SQL service endpoint on the pod’s management subnet when your pod uses subnets that you create yourself, and for allowing outbound access for port 5432.

At the September 2019 time, this High Availability (HA) feature for pods in Microsoft Azure is only supported for pods deployed in the Microsoft Azure Commercial regions (standard global regions). The pod HA feature is not currently supported for pods deployed in Microsoft Azure China, Microsoft Azure Germany, and Microsoft Azure Government (US Gov Virginia, US Gov Arizona, US Gov Texas). The VMware team is working on adding support for the HA feature for pods in those above listed cloud environments. If you have an existing pod in Microsoft Azure in China, Microsoft Azure Germany, or Microsoft Azure Government that you want to update to this release’s manifest version without HA, please contact your VMware representative for assistance.

Before updating an existing pod in one of the standard Microsoft Azure global regions to manifest 1600 or later, because the new pod architecture uses the Microsoft Azure Database for PostgreSQL service, you must ensure the following items are in place:

- If the pod is using a custom management subnet, you must add the Microsoft.SQL service endpoint to that management subnet prior to the pod update process. See the steps in When Using Existing Subnets for a Horizon Cloud Pod in Microsoft Azure.

- You must ensure your firewall rules and network security groups allow pod communication to the Microsoft Azure PostgreSQL database server over the management subnet. See the entry for TCP port 5432 in DNS Requirements for a Horizon Cloud Pod in Microsoft Azure.

- Ensure the pod’s associated subscription includes the resource providers that are listed in Step 8 of Create the Required Service Principal by Creating an Application Registration.

- To increase resiliency of the Horizon Agent pairing process, this release brings further evolution of moving the DaaS agent functions into the Horizon Agent. The DaaS agent has now been incorporated into the Horizon Agent. Even though both the automated Import Image workflow and manual installation of the Horizon Agents Installer install the DaaS
agent’s MSI into the guest operating system as they did in previous releases, starting in this release, the DaaS agent is dormant and not used. However, the service for the DaaS agent still appears in the Windows Services list. Do not start that service or unexpected results can occur.

- Moving the DaaS agent functions into the Horizon View agent has changed both the automated Image Import from the Azure Marketplace workflow and the steps for manually building a base VM. Previously, the base VM that resulted from the automated workflow was paired to the cloud at the end of the workflow, while for a manually created VM, you had to manually bootstrap and pair the VM. Now, for base VMs in a pod that is new or updated to this release version, the resulting base VM is listed on the Imported VMs page with an agent status of Not Paired. To pair the VM, you can either:
  - Run the **Reset Agent Pairing** action on the VM listed on the Imported VMs page, if you want to pair it with the cloud before customizing it.
  - Run the **New Image** action on the VM directly, if the VM has all of the customizations you want and you are ready to publish it. In this case, the New Image workflow will first run the pairing process to make the agent active, and then you can complete the rest of the fields and click **Publish** to publish the image.

- With the move of the DaaS agent functions into the Horizon View agent, the Reset Agent Pairing workflow is now available to use on imported VMs, farm server VMs, and desktop VMs in dedicated VDI desktop assignments. In a farm’s details or dedicated VDI desktop assignment’s details, if you see an error state in the Agent Status column for a farm server VM or desktop VM, you can use the **Reset Agent Pairing** action in the console to repair the pairing state of that VM. (The action is not available for floating VDI desktop assignments.) In the Imported VMs page, you can use the **Reset Agent Pairing** action to initially pair a VM that has not yet been paired, or repair the pairing state for a VM that was previously paired.

- The disk encryption feature now uses the newer AzureDiskEncryption v2.2. This newer version enables support of disk encryption for VMs with in-guest proxy set up to talk to the Internet. To take advantage of this new support, update your VMs' agents to version 19.3.0 or later.

- Updated guidance to use VM models that have a minimum of two (2) CPUs for your farms and VDI desktop assignments. VMware scale testing has shown that for production environments, using a minimum of 2 CPUs 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.

- Usability enhancements to the VM Types and Sizes page.

**Additional items of note for current customers**

- Improved usability and optimization of the Unified Dashboard interactive map view, including more accurate reflection of the pod location and zoom functionality.
- Enhancements in the reports available in the Administration Console's Reports page. The data in these reports is provided by the Cloud Monitoring Service.
- For cloud-connected Horizon 7 pods, additional details are displayed in a pod's details page. The pod and Cloud Connector must be at the latest version to see this feature.
- The product name formerly known as VMware User Environment Manager™ is now named VMware Dynamic Environment Manager™.

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 Horizon Cloud pod. Over time, this topic will be updated to list more of the known limits.

These supported maximums are the result of testing the service up to these maximums.

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 number of concurrent connected sessions per Horizon Cloud pod

A Horizon Cloud pod can support up to 2,000 active concurrent sessions. This number includes the individual connections to VDI single-session desktops using Windows client-type operating systems, to multi-session desktops using Windows 10 or 11 Enterprise multi-session or Windows 11 Enterprise multi-session, to multi-session desktops based on Windows Server operating systems, and RDS-based applications served by the pod.

As an example, when your pod has 400 active user sessions to 400 VDI single-session desktops, 1500 active user sessions to a Windows 10 or 11 multi-session machine, and 100 active user sessions to a Windows Server 2016 multi-session machine, that reaches the 2,000 maximum for the pod (400 + 1500 + 100 = 2,000).

Exceeding more than 2,000 active sessions per pod is not supported.
Also, when the pod is configured to use an Unified Access Gateway configuration, that 2,000 number is supported only when the F8s_v2 VM model is used for the Unified Access Gateway appliances.

- When the F8s_v2 VM model is used for the Unified Access Gateway appliances, the pod can support up to that 2,000 maximum. When you expect usage in your environment to exceed more than 1,000 active sessions on the pod, you should specify the F8s_v2 VM model in the gateway deployment wizard.

- When the A4_v2 VM model is used for the Unified Access Gateway appliances, the pod can support up to 1,000 active sessions only. This choice is sufficient only for proofs-of-concept (PoCs), pilots, or smaller environments where you know that you will not exceed 1,000 active sessions on the pod.

Note that resource utilization on the Unified Access Gateway appliances is dependent on the user activity within their session and could vary based on your user profiles and types of users.

**Maximum number of Horizon Cloud pods per VDI multi-cloud assignment**

The supported maximum number of Horizon Cloud pods in a VDI multi-cloud assignment is five (5). Using more than five increases the concurrent load on Universal Broker, which is the brokering technology configured in your tenant environment for use with VDI multi-cloud assignments. Increasing that concurrent load can lead to the end users encountering failures when they click on the assignment’s displayed tile in the client and the service attempts the operation to log in the user to the virtual desktop.

In addition to adhering to the five-pod maximum per VDI multi-cloud assignment, you can further reduce the likelihood of end users encountering failures at the point when they click on the assignment’s displayed tile in the client by including an additional desktop capacity of three percent (3%) in the VDI multi-cloud assignment. As an example, when you are defining a VDI multi-cloud assignment for provisioning 1,000 virtual desktops to 1,000 users, size the assignment for 1,030 desktops.

**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 fleet of cloud-connected pods can consist of Horizon Cloud pods in Microsoft Azure along with Horizon pods installed in the VMware SDDCs (software-defined data centers) that are supported for such pods. As a result, all of those cloud-connected pods must have line of sight to the same set of Active Directory domains. If your pod fleet already consists of cloud-connected Horizon pods and you are deploying your first Horizon Cloud 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.

**Horizon Cloud — Known Limitations**

So that you can have the best success throughout your journey using Horizon Cloud, keep these known limitations in mind.

**All Deployment Types**

- Every pod associated with your Horizon Cloud customer account and connected to Horizon Cloud must have line of sight to the same set of Active Directory domains and have one-way or two-way trust configured along with that line of sight.

- The system retrieves the data for the Utilization, Concurrency, Session History, and Top Applications reports once a day, at a specific UTC time. The data for the Utilization and Concurrency reports is retrieved at 2 AM UTC, the data for the Session History report is retrieved at 2:10 AM UTC, and the data for the Top Applications report is retrieved at 2:30 AM UTC. As a result, the reported information that is displayed in the administrative console might not reflect the data collected between the last retrieval time and the time at which you are viewing the reports in the console. As an example, because the logic for the Users and Peak Concurrency data in the Concurrency report is calculated on the daily basis for which the data is retrieved, the data from user activity on April 23 is calculated at the 2 AM UTC time point on April 24 (the following day). After that time point is passed and the system retrieves the collected data, the data from April 23 gets displayed in the report. If one of your end users starts a session after the 2 AM UTC time point on April 23, data for that user’s session will not be reflected in the on-screen report until after 2 AM UTC on April 24.

**Deployments - Horizon Cloud Pods in Microsoft Azure**

A Horizon Cloud pod is the type that is built on the Horizon Cloud pod-manager technology.

- Due to limitations with how Microsoft Azure VNets handle concurrent subnet creation and deletion operations, running concurrent pod-related operations that require modifying the same VNet at the same time can result in failure to complete those operations. To avoid
running into this issue, avoid running pod deployment, pod deletion, or pod edit operations that involve subnets at the same time when those pods are using the same VNet. Here are some examples of concurrent pod-related operations involving modifying the VNet where the chances of experiencing concurrent subnet actions on the VNet can occur, resulting in failures to complete the operations:

- When you do not create your subnets ahead of time and are having the pod deployer create the subnets using CIDRs, and you initiate creation of two pods concurrently on the same VNet. The subnets are being added to the VNet for both pod creations simultaneously.

- When one pod is deploying and you initiate deletion of another pod on the same VNet. The subnets are added to the VNet for the deploying pod at the same time the other pod’s subnets are being deleted from the same VNet.

- When you edit a pod to add an external gateway configuration in the pod’s VNet using CIDR blocks while another pod is in progress of being deleted. The subnets are added to the VNet for the gateway configuration at the same time the other pod’s subnets are being deleted from the VNet.

- Expanding the size of the pod’s subnets after the pod is deployed is not currently supported. Before you deploy a pod, you must ensure the address spaces for the subnets you specify in the deployment wizard are large enough to accommodate your expected usage.

**Note** To workaround this limitation, a new feature made available in pods of manifest 2298.0 and later provides for adding tenant subnets for use by your farms and VDI desktops assignments after the pod is deployed. This feature provides for flexibility to add tenant subnets located in the pod’s same VNet or in a peered VNet for use by your farm and desktop VMs after the pod is deployed. For more details, see the Administration Guide.

- Multiple pods cannot share the same fully qualified domain name that is set on their Unified Access Gateway configurations. Each pod configured with Unified Access Gateway instances needs its own unique fully qualified domain name (FQDN). The FQDN cannot contain underscores.

**Note** As of the December 15, 2020 cloud plane refresh, having different FQDNs for the pod’s external gateway and internal gateway configurations is supported. Prior to December 15, 2020, the system enforced having the same FQDN for a pod of manifest 2298 and later. Now it is your choice if you want the pod’s gateways to use different FQDNs or the same FQDN. If you want to use the same FQDN on both gateways, after pod deployment, you would 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.
- Editing or updating of the proxy settings on a pod after the pod is deployed in Microsoft Azure is currently unsupported. Also, adding a proxy configuration to a deployed pod that was deployed without proxy settings is currently unsupported.

- The NSX Cloud capabilities in this release are not supported for Microsoft Windows Server 2019.

- When you deploy a Horizon Cloud pod in Microsoft Azure after you have already configured True SSO for previously deployed pods, the system does not automatically pair the new pod with the Enrollment servers. You must manually repeat the steps to export the pairing bundle and import it into the Enrollment servers. For the steps, see the topic Configure True SSO for Use with Your Horizon Cloud Environment and its subtopics.

- In workflows that result in the system creating VMs, such as creating farms, images, and assignments, if you try to enter a name that is longer than length supported by the system for the to-be-created item, the system prevents you from entering more than the supported number of characters. The number of characters supported for an item's name depends on the workflow.

- In a Microsoft Azure multi-pod environment, you cannot reuse names that you used in one pod when creating items in another pod. The reason for this limitation is that pods in the multiple-pod environment share the same Active Directory domain and the same VNet. As a result, if names are shared within such multiple-pod environments, unexpected behavior can occur. This limitation applies to names for image, farms, and VDI desktop assignments. Ensure that unique names are used for your images, farms, and VDI desktop assignments.

- Follow these rules when entering characters in the administrative console:
  - Use only standard ASCII characters in user names and passwords, and for the password when downloading the DaaS SSL bootstrap file. If you use non-ASCII characters for these items, unexpected results might occur.
  - When entering names for imported images, farms, assignments, and other assets that result in creating a VM in Microsoft Azure, do not enter more than 12 characters for the name.
  - Do not use commas in user passwords.
  - When using the Import VM wizard to create a base image VM from the Microsoft Azure Marketplace:
    - Enter a user name and password that adheres to the Microsoft Azure requirements for VM admin user names and passwords. See the Microsoft Azure FAQ page for details.
    - Do not enter a name for the image that ends with a hyphen (-).
    - Do not include an underscore character (_) in the image name.
Updates of Horizon Cloud Pods in Microsoft Azure

- When the Horizon Infrastructure Monitoring feature is enabled on a pod and the pod is updated via the blue-green update process, the status of the blue pod manager VMs displays as red or inactive for up to 12-14 hours in the Infrastructure dashboard. After 12-14 hours, those blue pod manager VMs are cleared from the Infrastructure dashboard automatically.

- During the process of updating a pod from an earlier software level to the latest one, end users who have connected sessions to the updating node will have those active sessions disconnected. No data loss will occur – except for the case where the RDSH farm or VDI desktop assignment serving the sessions has the Logoff Disconnected Sessions set to Immediately. For such farms and VDI desktop assignments, the disconnected sessions are also logged off immediately and in-progress user work is lost in those conditions. After the update process is complete, those users can reconnect. The pod update process typically takes less than a half hour. However, some pod updates might take longer.

- When pods running on manifests prior to the October 2020 release's 2474 manifest are then updated to 2474 or later, if you had used the Microsoft Azure Portal to manually add tags directly on resources or resource groups created by Horizon Cloud in your subscription — such as creating custom tags on the farm or VDI desktop assignments' resource groups — when those pods are then updated, the pod update process does not preserve those custom tags that you had directly added using the Microsoft Azure Portal. Those custom tags will be removed. After the pod update, you must subsequently use the Horizon Universal Console features to edit the farms and VDI desktop assignments to have the system apply those tags on the resource groups for those farms and VDI desktop assignments. Using the console's Azure Resource Tags feature is the supported method of adding resource tags on the pod-created resource groups for farms and VDI desktop assignments. In each of the following documentation topics, read the descriptions of Azure Resource Tags fields located in them. The same fields are used when editing a farm or VDI desktop assignment.

  - Create a Farm
  - Create a Dedicated VDI Desktop Assignment Provisioned by a Single Pod in Microsoft Azure
  - Create a Floating VDI Desktop Assignment Provisioned by a Single Pod in Microsoft Azure
  - Horizon Cloud Pods in Microsoft Azure - Create a VDI Multi-Cloud Assignment in Your Horizon Cloud Tenant Environment

Imported VMs, Golden Images, Farms, or VDI Desktop Assignments - Horizon Cloud Pods in Microsoft Azure

- Although the console does not prevent you from using images obtained from origins other than the Azure Marketplace in the console's workflows, use of such images is unsupported.
To be supported for use in Horizon Cloud on Microsoft Azure, all imported base images must be built from Windows-based VMs that are sourced from the Azure Marketplace. Even if you try an image obtained from other origins and the console does not prevent you from using it within the console's workflows, use of such images is unsupported.

Additionally, if a Windows 11 image, it must be sourced directly from the Azure Marketplace and cannot have been subsequently processed. Importing Windows 11 VM from any other sources such as Shared Image Gallery (SIG), Azure Managed Images, Azure VM snapshot, and the like is currently unsupported.

- Generation 2 Azure VMs are currently supported for use only with Windows 11 single-session operating systems and with Windows 11 Enterprise multi-session operating systems.

- Currently, the Azure GPU-capable NVv4 VMs that use the AMD Radeon Instinct graphics drivers are supported for use only when they are imported using the custom import method. The custom import method is also referred to as a manual import in this documentation. The automated Import Virtual Machine from Marketplace wizard does not provide this feature currently.

- Also, the service does not currently support use of Windows 11 with these NVv4 VMs and the AMD Radeon Instinct graphics drivers. That use has not been qualified.

- The service's support for Windows 11 has some known considerations, limitations, and issues. For those details, see Support for Windows 11 Guest Operating System - Considerations, Known Limitations, and Known Issues.

- Use of True SSO with session desktops, remote applications, or VDI desktops based on images running Windows 10 version 2004, Windows 10 version 20H2, Windows Server version 2004, or Windows Server version 20H2 requires installation of a Microsoft patch on those operating systems. The patch fixes a Microsoft issue that blocks True SSO from authenticating with those operating systems. For details, see VMware KB 79644 which references Microsoft Update KB4598291.

- Use of the disk encryption feature for farms and VDI desktop assignments is not currently supported for pods in Microsoft Azure Government clouds.

- Currently, use of the following Horizon Agent features is not supported: VMware Logon Monitor service. By default, the Horizon Agents Installer deactivates the VMware Logon Monitor service in all installations that the installer performs.

- The USB redirection capability is not supported when using the VMware Horizon Client for Android to access virtual desktops and remote applications served by your Horizon Cloud environment.

- For GPU-enabled golden images based on server-type operating systems, Microsoft Windows Server versions 2016 and 2019 are recommended to avoid limiting the number of end user sessions. Due to an NVIDIA driver limitation on Windows Server 2012 R2, the maximum number of sessions for each RDS desktop server is 20.
If you have an image using Microsoft Windows 10 1709 (RS3) and you want to update it to Windows 10 1803 (RS4) or Windows 10 1809 (RS5), first upgrade that Windows 10 1709 to the latest Horizon Agent version 19.4 before proceeding with upgrading the Windows operating system.

By default, when you use the automated Import Virtual Machine from Marketplace wizard to create an image with a Windows 2012 server operating system, the resulting image does not have the Desktop Experience enabled. If you want the resulting image to have the Desktop Experience, you must manually enable the Desktop Experience in the resulting image.

If you initiate converting a desktop to an image but cancel before the task finishes, a second attempt to convert the desktop to an image may fail. To avoid this issue, you should power off the desktop and power it on again before attempting to convert it to an image a second time.

In a URL redirection customization, URL patterns are treated as case sensitive when they are intercepted by the Horizon Client. For example, URL redirection does not occur for URL patterns specified as *GOOGLE.com and *Google.com, even though the pattern *google.com is redirected. Redirection for the end users does not occur if the specified pattern does not match the actual character case used in the target file systems.

### Deployments - Horizon Pods

A Horizon pod is the type that is based on the Horizon Connection Server software. For background information on the various deployment architectures used for Horizon pods, see Horizon Pod Deployment Architectures. For detailed information about the various cloud-plane services, see the administration guide.

The following limitations apply in the current release and are updated as applicable for each Horizon Cloud Service release.

- The automated update feature of Horizon Cloud Connector is only supported for Horizon pods deployed on premises. To update a Horizon Cloud Connector instance that is paired with a Horizon pod deployed in a cloud environment, follow the instructions in Manually Update the Horizon Cloud Connector Virtual Appliance.

- The Horizon Image Management Service (IMS) is currently supported for use only with a subset of the Horizon deployment models that are described in VMware Tech Zone’s Horizon Reference Architecture. For information about the specific deployment models currently supported by IMS, refer to IMS System Requirements. As more models from the Horizon Reference Architecture are qualified for IMS support, the newly supported ones are added to the list.

### Workspace ONE Hub Services and Universal Broker - Integration

- This feature is available for VMware Workspace ONE® Access™ Cloud only. It is not supported for VMware Workspace ONE Access on premises.
With this integration, end users can access their Horizon Cloud desktops and applications from the Hub catalog. Ensure that you configure the required settings for VMware Workspace ONE® Intelligent Hub, as described in Workspace ONE Access - Configure Intelligent Hub for Integration with Horizon Cloud.

In this release, this integration supports end-user access using these clients: browser-based Hub catalog, Workspace ONE Intelligent Hub for Windows, and Workspace ONE Intelligent Hub for macOS. The minimum version of the Windows and macOS desktop apps required for this support is 21.05.

Password caching is not turned on by default in new Workspace ONE Access tenants. If True SSO is not enabled in your Horizon environment, you can enable password caching to cache users’ passwords so that they are not required to enter their passwords again while starting Horizon Cloud desktops and applications. For more information, see Configuring Password Caching for Virtual Apps (Workspace ONE Access Cloud Only).

Access policies set in Workspace ONE Access do not apply to applications and desktops from a Horizon Cloud environment that has Universal Broker enabled.

**Horizon Universal Console - Related Caveats and Limitations**

- The console does not fetch the currently effective names of your Active Directory users and groups which are already specified in an assignment's details. When you change the name of a user or group in your Active Directory, the console continues to display the prior name of the user or group in the assignment's details from when that user or group was initially added to the assignment. While editing an assignment and searching for a user or group in the search field, the current effective names of the users or groups are displayed in the console. However, even after saving the updated assignment, you will continue to see the old initial name displayed in the assignment's details. There is no functional impact of this limitation.

- The Web-based administrative console is not supported in the Apple Safari browser. Some user interface features might not work correctly. In a Mac OS, instead of Apple Safari, you can use Chrome or Firefox browsers.

- Your authenticated (logged in) session into the console will time out after the time setting that is configured in the console's General Settings page. The default is 30 minutes. If you have at least one cloud-connected pod, you can change the default to a value ranging from 30 minutes to 180 minutes. In most cases, when the configured time is up, the system will automatically explicitly log you out and present a message that you must log back in. However, sometimes the system ends your authenticated session and does not explicitly log you out. When that happens, when performing certain tasks in the console, error messages might be displayed which do not accurately reflect the current state, such as the node deployment wizard fails to validate your subscription entries, values are not displayed in drop-down menus, and the Farms page reports no node is available in which to create a farm and error messages stating "No service _sessions of type identity_node were provided". If you start to see such behavior and you have been using the console for 30 minutes or more, manually log out and then log back in.
Horizon Infrastructure Monitoring - Related Caveats and Limitations

- Horizon Infrastructure Monitoring is currently unsupported for use with Horizon Cloud on Microsoft Azure pods in Azure China regions because access to the required packages.cloud.google.com site is disallowed from Azure China regions at this time.

Horizon Cloud — Known Issues

This topic lists known issues that you might encounter when using the service and known workarounds, if any.

This documentation topic includes the known issues for Horizon Cloud Connector. However, even though you use the Horizon Cloud Connector to connect Horizon pods to Horizon Cloud, for the known issues for the software running inside those Horizon pods, refer to the release notes at the following locations, according to the pod's Connection Server software version:

- Version 7.13 - available at Horizon 7 Documentation.
- VMware Horizon 8 versions - available at Horizon Documentation.

For Image Management Service (IMS) known issues involving IMS features that are available in production for all Horizon Cloud customers, see the known issues page in Management Horizon Images from the Cloud.

**Note** The numbers in parentheses stated in each known issue refer to VMware internal issues tracking systems.

Login Related Known Issues

Even though you successfully created a password for your My VMware account that contains a backslash (\), logging in to Horizon Cloud using those credentials fails (2595757)

When you use My VMware credentials to login to Horizon Cloud, passwords which contain a backslash are not supported. To see the list of supported special characters, log in to my.vmware.com and navigate to your profile's Change Password section. That page will display the supported special characters. Workaround: Reset your My VMware account password to a new one and ensure the new one does not contain a backslash (\).

Active Directory Related Known Issues

Primary bind account lockout is not detected until you perform an action involving Active Directory in the administrative console. (2010669)

Due to this issue, an administrator logged into the Web-based administrative console will not see a primary bind account lockout notification until an action involving Active Directory is performed in the user interface, such as when searching Active Directory to add users to assignments. The underlying services only detect a locked-out service account when they
make a request to talk to Active Directory for either authenticating or searching (user or group). Workaround: None.

It takes up to 15 minutes for the Web-based administrative console to reflect a lockout or unlocked state of the primary bind domain account. (2009434)

The system's connection object to Active Directory is cached for 15 minutes. As a result, it might take 15 minutes from the time point when the primary bind account goes to locked state and the system raises the notification to the administrator. Conversely, after the administrator clears the locked-out condition of the account, it might take up to 15 minutes for the system to stop notifying about the now-cleared account. Workaround: None.

For farms in a pod in Microsoft Azure, reusing the same farm name with a different domain in the same Active Directory forest can lead to domain join failures due to duplicate service provider names (SPNs). (1969172)

Due to a new feature for domain controllers in Microsoft Windows Server 2012 R2 and higher, a duplicate SPN check on the domain controller causes domain join failures. See the Microsoft KB article 3070083. Workarounds:

- Avoid reusing farm names.
- As described in that Microsoft KB article, turn off duplicate SPN checks in the Active Directory domain.

When using Azure AD Domain Services, the Active Directory registration workflow fails at the domain join step with an error that the Reset Password permission is lacking. (2218180)

The Horizon Cloud team has verified that adding the required domain join account permissions works the same when using Azure Active Directory (AD) Domain Services with your pod as for other Active Directory domain deployments. See the Microsoft documentation topic Create an Organizational Unit (OU) on an Azure AD Domain Services managed domain that describes the built-in container AADDC Computers and see also the Important note at the beginning of that topic about enabling password hash synchronization to Azure AD Domain Services. Before setting the permissions on the domain join service account, it is important that you follow the Microsoft documentation about enabling password hash synchronization to Azure AD Domain Services for the domain join service accounts. If you continue to experience a domain join permissions error in the Register Active Directory workflow after following the Microsoft documentation, please contact VMware support and reference problem report number 2218180.

Microsoft Azure Subscription Related Known Issues

After using the Horizon Universal Console to update the secret key for a pod’s Azure subscription settings, the pod manager VMs must be restarted for the new credentials to take effect (2979394, 3007687)

Due to this known issue, after you edit and save the Application Key setting in the console's Manage Subscription window, the newly entered secret key does not take effect on the pod...
manager VMs until the management service is restarted in the VMs' operating systems. If the management service is not restarted, API calls used by the service to work with resources in the subscription will begin failing. Workaround: If your pod is in this situation where its subscription secret key needs updating for some reason, such as an approaching or past expiration date, please open a service request for assistance from VMware Support and Horizon Cloud Operations teams to ensure the sequence of steps is performed successfully. At a high-level, the steps are:

1. In the Azure Portal, generate a new secret key.

2. In the Horizon Universal Console, follow the standard steps to update the secret key that is used by the pod or pods associated with the old key, as described in the page Change, Modify, and Update the Subscription Information Associated with Deployed Horizon Cloud Pods.

3. Ask VMware Support to perform the restart of the management service in both pod manager VMs.

The specific command to restart the management service is not suitable for public posting here because only the VMware teams will be able to run that command. Those teams can reference internal issue 3007687-update-9.

Cloud Connector Related Known Issues

The no-proxy host configuration specified in the No Proxy For field when deploying the OVF template is not saved to the deployed appliance (2454245, 2466306, 2467017, DPM-5388)

This issue is resolved in Horizon Cloud Connector Version 1.6 or later. When running the Deploy OVF Template workflow in your vSphere environment, you have the option to specify a no-proxy host configuration in the No Proxy For field. However, due to this known issue, the entered settings do not get captured in the deployed appliance's configuration files. As a result, the deployed appliance does not honor the specified no-proxy host setting.

Horizon Cloud Pods' Gateway Configuration Related Known Issues

The Edit Pod feature for enabling syslog server settings in the gateway configuration is not updating the Unified Access Gateway configuration with those settings. (3005985, 3023935, 3026855)

After you edit a pod's gateway settings to enable the syslog server feature in the Edit Pod wizard and save the changes, the console indicates that the settings are saved successfully to the system. However, due to this issue, the system's API call to update the Unified Access Gateway configuration fails to complete. As a result, the syslog server setting is not actually enabled in the Unified Access Gateway configuration, with the effect that the Unified Access
Gateway instances do not generate the expected syslog traffic. Workaround: Not currently available. This entry will be updated when a workaround is available or this issue is resolved.

Universal Broker Related Known Issues

The Horizon Universal Broker client on the Horizon Cloud Connector does not consume proxy-related updates that you make in the connector appliance after the appliance is initially deployed (HD-35551)

This issue is resolved in Horizon Cloud Connector Version 1.6 or later. The Horizon Universal Broker client in the connector appliance picks up proxy details during the first-time boot of the appliance. Because the first-time boot runs only during the very first time the appliance is powered on after deploying the OVF template, any subsequent changes to the appliance's proxy configuration settings are not consumed by the Horizon Universal Broker client. Taking this known issue together with the above known issue about the no-proxy configuration during OVF template deployment means that any hosts related to the Horizon Universal Broker cannot be set as non-proxy hosts.

Error message 'Failed to connect to the Connection Server' is displayed when your Horizon Client or Horizon HTML Access in the browser starts to connect to the Universal Broker (2714266)

This issue affects Horizon Cloud pods in Microsoft Azure that are running manifest 2632.x, in tenants configured to use Universal Broker for brokering end user desktops. Another manifestation of this issue is you observe both of the following occurring at the same time:

- In the pod details page for the pod where the desktop VM resides, you see the pod manager VM health is reported as Error for all of the pod's pod manager VMs.
- Error message 'This desktop is currently not available. Please try connecting to this desktop again later, or contact your system administrator.' is displayed when the Horizon Client or Horizon HTML Access in the browser starts to connect to the Universal Broker.

This issue might occur intermittently for Horizon Cloud pods in Microsoft Azure that are running manifest 2632.x if the pod manager instance was rebooted. Sometimes when the pod manager instance has been rebooted (a rare, atypical situation), after the pod manager instance is powered up and the Universal Broker attempts a connection to the pod, it cannot make an authenticated connection. Workaround: None. If you encounter this situation, open a service request for assistance and cite the internal problem report number 2714266.

For pods at manifest 2747.x and later, this issue is resolved.

Images, Farms, Assignments Related Known Issues

The known issues listed here apply to pods deployed in Microsoft Azure.

During the image publishing process, a timeout error occurs and the VM remains powered on, and prevents the publish flow from successfully completing (2954270, 2962049)
This issue is the result of an issue in the Microsoft Azure hypervisor that occurs when running the sysprep step of the publishing process. The issue occurs in some Azure VM models. For additional details, refer to VMware Knowledge Base article KB88343.

Based on the Microsoft Azure team’s recommendation, to provide a resolution for Horizon Cloud customers, the default Azure VM model used by the service’s automated Import VM from Marketplace wizard is changed in the service’s v2204 release to use the Standard_DS2_v2 model for the automated import of non-GPU Windows 10 VMs (both single-session and multi-session):

- For single-pod images, the automation’s default VM model is changed from the previously used Standard_D4_v3 VM model to use Standard_DS2_v2.
- For multi-pod images, the automation’s default VM model is changed from the previously used Standard_D2_v2 model to use Standard_DS2_v2.

As of the v2204 release, please include quota for the Azure DSv2-series in your pod’s Azure subscriptions.

VMs and their related resources might not delete completely in the Microsoft Azure subscription. (2824239, 2681761, 2750176)

This issue is resolved for pod manifests 2915.x or later. If or when this issue occurs in pods of earlier manifests, issues such as problems expanding VDI assignments can occur. This issue is due to an issue in Microsoft’s Azure Resource Manager (ARM) and delays in replicating the resources’ status across the multiple regions of the Microsoft Azure cloud. Due to this Microsoft ARM issue, some of those VM-related resources might be left undeleted, and therefore unattached to a VM in the Azure subscription. Examples of such unattached items that might occur are disks and NICs. Workaround: This issue is resolved for pods running manifests 2915.x or later. If you encounter this issue, please file a service request (SR) to request assistance with clearing the stale data and to schedule your pod upgrade to prevent recurrence of the issue. See KB article 2006985 for SR filing steps.

For pods deployed into Microsoft Azure Government cloud subscriptions, using the disk encryption feature in farms and desktop assignments fails. (2572579)

When your pod is located in Microsoft Azure Government clouds and you try to create a farm or VDI assignment with the disk encryption feature selected, the creation process fails with error Azure error encrypting the VM. Workaround: None.

In the Servers tab for an existing farm, all of the User Login Mode choices give an error message that the Horizon Agent must be updated. (2528295)

Use of the administrative console to set the User Login Mode depends on detecting agent version 20.1.0 running in the farm VM. However, that version of the agent might not yet be available in the cloud control plane for you to use to update the agents in your existing farm VMs. Workaround: None. When the 20.1.0 version of the agent is available in the cloud plane and your pods are updated to the manifest version that can consume that agent version, then you can update the farm VMs to that agent to use the User Login mode choices.
Sometimes some desktop VMs out of a large floating VDI desktop assignment report unknown agent status. (DPM-3201)

In floating VDI desktop assignments with large numbers of desktop VMs, due to a known issue, a small number of those desktop VMs can go into an unknown agent state because some Windows services, like the Horizon Agent’s Blast service or the Microsoft Azure service, do not start or are slow to start. As a result, in the administrative console, the Agent Status column for those desktop VMs shows "Unknown" state, with reported agent errors.

Workaround: In the console, use the **Restart** action to restart those VMs.

The Import VM from Marketplace wizard creates Windows Server 2012 images without the Desktop Experience enabled. (2101856)

Due to a known issue, when you use the automated Import VM from Marketplace wizard to create an image with a Windows Server 2012 operating system, the resulting image does not have the Desktop Experience enabled. Workaround: If you want the resulting image to have the Desktop Experience, you must manually enable the Desktop Experience in the resulting image. Note also that for the Windows Server 2012 operating system, to install the Horizon Agent with the Scanner Redirection option requires the Desktop Experience be enabled in the operating system.

When publishing (also known as sealing) an imported VM, the process might result in a timeout or other failures to publish due to sysprep failures. (2036082, 2080101, 2120508, 2118047)

After you click **Convert to Desktop** on an imported VM and **Publish** to make it a published (sealed) image, a number of operations are performed on the VM. These operations include running the Windows System Preparation (sysprep) process, shutting down the VM and powering it off, and so on. Due to industry-known issues with the Windows sysprep process and customizing virtual machines, sometimes the publishing process fails for various reasons. On the Activity page, you see messages like "Timeout Error Waited 20 minutes for virtual machine to power off.", and other sysprep failure message.

Generally speaking, you can avoid such sysprep issues when you create the VM using the Import Virtual Machine from Marketplace wizard and select **Yes** for the wizard’s **Optimize Windows Image** toggle. If you are seeing this error for an imported VM in which you did not use that option, or if you manually created that VM, refer to Microsoft KB 2769827, Microsoft MVP article 615 for best practices in configuring your image VM to minimize likelihood of having sysprep issues when you go to publish the image. If you continue to get sysprep issues, see the information in the articles Deciding to Optimize the Windows Image When Using the Import Virtual Machine from Marketplace Wizard and Using the Remove Windows Store Apps Option When Using the Import Desktop Wizard for ways that the automated Import VM from Marketplace wizard uses to reduce the changes of sysprep issues. If you see the timeout errors in the Activity page, you can try this workaround: on the Images page, use the **Convert Image to Desktop** action on the image. When the Activity page indicates converting the image to a desktop is successful, navigate to the Imported VMs page. Connect to the VM and apply the best practices described in the KBs. After you see the Imported VMs page reports the VM is powered on, select the VM and click **Convert to Image** to run the publishing process again.
During farm creation, sometimes the server VMs are stuck at the customization step. (2010914, 2041909)

Sometimes during the sysprep process on the farm's server VMs, a Windows service named tiledatamodelsvc prevents sysprep from accessing Windows files that it needs to complete the sysprep customization process. As a result, the farm's server VMs do not move past the customization step. The sysprep error log contains the line "Error SYSPRP setupdigetclassdevs failed with error 0". Workaround: If you encounter this issue and see that error message in the sysprep error log file, try stopping and deactivating the tiledatamodelsvc service in the image. Then create the farm.

Agent status might display as 'undefined' on the Imported VMs page after duplicating an image or manually creating an image in Microsoft Azure. (2002798)

When you use the Duplicate button on the Images page to clone a published image or when you manually create an image VM in Microsoft Azure, the resulting VM is listed on the Imported VMs page. Due to this issue, even when the VM is fully powered-on, the agent status might be displayed as 'undefined'. However, when you select the VM and choose Convert to Image to publish it, the user interface reports the agent in 'Active' state. Workaround: None. If the Reset Agent Pairing or New Image or Convert to Image workflows report the agent as 'Active', you can ignore the 'undefined' status on the Imported VMs page.

App Volumes on Pods in Microsoft Azure Related Known Issues

The known issues listed here apply to pods deployed in Microsoft Azure.

Uploading application packages (.vhd files) with the same file name to the same location (file share), captured at different times, can prevent the App Volumes service from attaching applications to the VDI Desktops when the user logs in (2783560)

Every time App Volumes captures an application package (.vhd file), the system generates a unique GUID to identify the volume or capture session. When you try to upload an application package to the staging fileshare of Horizon Cloud Azure pods with a (.vhd) filename that was uploaded previously, a mismatch occurs between the GUIDs already present in the Horizon Cloud Azure pods and the cloud services.

The App Volumes Manager service running on the Horizon Cloud Azure pods periodically imports the application packages from the fileshare. When you try to import the applications from the import Inventory > Applications page of the Horizon Universal Console, newly imported application packages and their corresponding GUIDs are mismatched with the GUIDs present in the App Volumes Manager service running the Horizon Cloud Azure pods. Because of this mismatch, assigned applications do not attach to the entitled users.

Removing some users or groups from an App Volumes assignment in the console might remove the entitlements from some of the remaining users or groups in the assignment (2704889)

Due to this issue, in the scenario where you have created an App Volumes assignment that contains a set of applications and specified users or groups, and then you edit that assignment...
and remove some of those specific users or groups, some of the users and groups that remain configured in that assignment find they do not see the applications in their entitled desktops.

Although this issue is resolved in pod manifest 2747 and later, you might encounter this issue in pods of earlier manifest versions. If you encounter this issue, you can work around it by creating a new App Volumes assignment with the required applications and users and groups, and delete the previously created App Volumes assignment.

When your environment has multiple pods in Microsoft Azure, the Capture process can sometimes go into an unknown state after the process has completed. (2600573)

When your environment has multiple pods with which you are using App Volumes, sometimes after running the capturing process, the console indicates the capture is in an unknown state even though the capture process on the VM has completed. To work around this issue, re-import the application package using Inventory > Applications > New > Import. As a result, the application package is successfully imported as a separate application and the subsequent assignment and application launch works.

In a Microsoft Windows 10 Enterprise multi-session deployment, a print job might terminate when another user logs in to the same machine.

In this environment, when a user with a app package assignment containing a printer driver logs in for the first time, an ongoing print job for another user on that multi-session machine might enter an error state. To work around this issue, wait a few minutes, or longer, after the print job terminates and attempt the print job again. See the Administration of Your Horizon Cloud Tenant Environment and Your Fleet of Onboarded Pods guide for related best-practices information.

In a Microsoft Windows 10 Enterprise multi-session deployment, users not assigned an app package receive aspects of the application

In this environment, occasionally, when you do not turn off auto updates for an application during provisioning, the updated portion of the app inadvertently becomes visible to all users (not just those assigned the application) on the multi-session desktop, such as in the form of a desktop shortcut and application binaries. To work around this issue, for applications with auto update services, add the application service name to the multi-string svservice registry configuration DisableAppServicesList to ensure that the auto update services are not started. See the Administration of Your Horizon Cloud Tenant Environment and Your Fleet of Onboarded Pods guide for related best-practices information.

Agent Update Related Known Issues

The known issues listed here apply to pods deployed in Microsoft Azure.

When you attempt an agent update on an image that has a Windows update pending, the update process might fail. (2234964)

If the image needs an update to the Windows OS, as opposed to a minor non-OS update, this can cause OS resources to be offline and not available for the agent update. Workaround:
Wait until the Windows update is complete and retry the agent update. To confirm that all Windows updates are complete, you can take the image offline, perform all pending updates, and re-publish the image before initiating the agent update.

### Reports and Monitoring Related Known Issues

The known issues listed here apply to pods deployed in Microsoft Azure.

**In the User Activity report, the displayed weekly average (hrs) is not intuitive. (1817065)**

Due to this issue, the weekly statistics fluctuate along the time because the calculation logic is dividing the current week's duration by seven (7) and not rounding up to a whole week. For example, when you select the last 30 days, the data for completed weeks is unchanged but the data for the current week is divided by seven (7). The current logic is weekly average (hrs) = daily average (hrs) \(*\) 7 days, resulting in the last 30 days weekly average = (total duration / 30 days) \(*\) 7 days. Workaround: None

**The Desktop Health report does not reflect a newly updated farm or VDI desktop assignment name until an hour after the name change. (1756889)**

If you change a farm's name or VDI desktop assignment's name, it takes an hour for the Desktop Health report's Assignment drop-down menu and Assignment column to reflect the new name. Workaround: Wait an hour before expecting the new name to appear in the report.

**The formatting in some of the CSV files that you can export from the Reports user-interface screens do not match the on-screen tables. (2015500)**

Some of the Reports page's subscreens provide an export feature to export the displayed data in CSV format. Due to this issue, the formatting in the CSV files exported from the Desktop Health, Concurrency, and Session History reports do not precisely match the ones you see displayed on the screen. For example, the column headings might be different and the CSV files might have more columns of data than in the on-screen tables. Workaround: None.

### Identity Management, Workspace ONE Access, True SSO Related Known Issues

The known issues listed here apply to pods deployed in Microsoft Azure.

**When a pod of manifest versions prior to 1763 is updated to manifest 1763 or later, and that pod has two-factor RADIUS configured on its Unified Access Gateway instances and is also integrated with Workspace ONE Access, you see that launching a desktop from Workspace ONE Access using the browser will display the RADIUS login form with the user name field prefilled with the user's UPN. (2248160)**

This symptom occurs because of a change that was released in VMware Horizon HTML Access 4.10. When your pod in Microsoft Azure from a previous Horizon Cloud release is configured with Unified Access Gateway instances and two-factor RADIUS authentication and you configure that pod to use Workspace ONE Access, previously when launching a desktop from Workspace ONE Access using the browser, the RADIUS login form prompts for the
user name and passcode. The end user would type the user name and passcode in the form. However, due to this issue, after upgrading that pod to this release, using the same desktop launch steps, the RADIUS login form has the user name field prefilled with the domain user’s UPN. This behavior only occurs when using the browser to launch the desktop. It does not occur when using Horizon Client. Workaround: If this situation is encountered, the end user can clear the prefilled user name field and enter their information. Generally, for most environments that are integrated with Workspace ONE Access, the two-factor authentication would be configured in Workspace ONE Access and not on the underlying Unified Access Gateway instances, in which case this issue would not be encountered.

Launching a second desktop from Workspace ONE Access using the Horizon Client can fail with the error 'You are not entitled to that desktop or application'. (1813881, 2201599)

This symptom occurs in the following situation. The user has entitlements to two dedicated VDI assignments through a group entitlement. Both dedicated VDI desktop assignments are listed in Workspace ONE Access when the user logs in. The user launches the first desktop using Horizon Client. That desktop connects. Then the user tries to launch the other desktop from the other assignment, also using the Horizon Client. The launch of that other desktop fails with an error indicating the user is not entitled. However, this issue is seen only for the first attempt on the second desktop. If the user launches the second desktop using the browser, subsequent attempts to launch the second desktop using Horizon Client succeed. Workaround: If you encounter this situation, try launching the second desktop using the browser.

Workspace ONE Access does not display the remote applications’ display names that you set in the Horizon Cloud administrative console. (2131583)

This issue is resolved by using Workspace ONE Access Connector version 19.03. Due to a known issue in versions of Workspace ONE Access Connector prior to 19.03, when Workspace ONE Access displays the remote applications that you sync from Horizon Cloud, Workspace ONE Access does not display the display names that you set for those remote applications in Horizon Cloud. Even though Horizon Cloud sends the display names to Workspace ONE Access, Workspace ONE Access uses the remote applications’ launchIDs instead. As a result, Workspace ONE Access displays the basic names for the remote applications.

User Interface Related Known Issues

Unless otherwise noted in the known issue text, the known issues listed here apply to pods deployed in Microsoft Azure.

The Logon Segments chart displayed in the session dashboard has no data.

This issue applies to all types of pods. The VMware Logon Monitor service provides the data for the Logon Segments chart that appears in the session dashboard. However, this release does not support use of the VMware Logon Monitor service and by default, the Horizon Agents Installer deactivates the VMware Logon Monitor service in all installations that the installer performs. As a result, even though no data is reported that the Logon Segments
chart can display, you see the Logon Segments chart is still visible in the session dashboard.

Workaround: None.

When using the administrative console in one browser tab, if you try to launch a disconnected desktop that you have in another browser tab in the same browser, the HTML Access portal is also logged off and you must log back in to the HTML Access portal itself. (2118293)

Usually when you launch a desktop and disconnect from it without logging out of the desktop, you stay logged in to the HTML Access portal itself and you can reconnect to the disconnected desktop without having to enter credentials to the HTML Access portal. Due to this issue, if you are in a browser window where you are logged in to the console in one browser tab and use another browser tab to log in to the HTML Access portal and launch a desktop, when you disconnect from that desktop and try to reconnect to it, the HTML Access portal logs off. Then you must re-enter credentials to the HTML Access portal before you can reconnect to that desktop. Workaround: To avoid this issue, log in to the administrative console using a separate browser window from where you have the HTML Access portal. This behavior only occurs if you are also logged in to the console in a browser tab in the same browser window in which you are also using the HTML Access portal.

In the User Card screen for a specific user, VDI dedicated desktop assignments are removed from the Assignments tab after the user's first launch of the dedicated desktop from that assignment. (1958046)

When a user is specified in a VDI dedicated desktop assignment as an individual user, not through an Active Directory group, that VDI dedicated desktop assignment appears in the Assignments tab in the User Card screen for that user only until the user's first launch of a dedicated desktop from that assignment. After the user's first launch of a VDI dedicated desktop from that assignment, the user card's Assignments tab no longer displays that VDI dedicated desktop assignment for that user. The user's first launch results in that user claiming a specific dedicated desktop from the underlying pool defined by that assignment and the system maps that specific dedicated desktop to that particular user. When that mapping is made, that specific dedicated desktop gets the Assigned state, and it is listed on the user card's Desktops tab for that user.

Workaround: Instead of relying on the user card's Assignments tab in this case, to see the already launched VDI dedicated desktops assigned to a specific user, you can use the Desktops tab. If you need to locate the specific VDI dedicated desktop assignment in which that user-desktop mapping is made, obtain the desktop name from the user card's Desktop tab and use the search by VMs feature of the top banner search to list that specific desktop VM. In the results from the search by VMs, click the name to open the specific assignment page that has that particular dedicated desktop. Then you can locate the user in the assignment's details.

The What's New screen appears even though you previously selected the option not to continue showing it. (2075825)

This issue applies to environments with any pod type. Due to this issue, if you clear your browser cache or you use a different browser than the one in which you previously selected
the option to not show the What's New screen, the screen might appear when you log in to
the administrative console. The flag for whether to show the What's New screen is stored in
the browser's local cache, instead of per user. Workaround: None.

Even though the image creation process has not fully completed, the Getting Started screen
displays Completed for the Create Image step. (2100467)

Due to this issue, the Create Image step is marked as completed prematurely. Workaround:
Use the Activity page to verify that the image creation process has completed.

When using the administrative console, you might see placeholders instead of the actual text
strings or you click a button on a page and nothing happens. (2045967)

This issue applies to environments with any pod type. VMware periodically updates the in-
cloud management environment that hosts the Web-based console. This issue can occur when
static content has been cached in the browser prior to the latest in-cloud update. It is a
temporary issue that will clear when the browser cache is cleared. Workaround: Try logging
out of the console, clearing the browser cache, restarting the browser, and then logging back
in to the console.

Application names are displayed in lowercase characters when end users access them using
Workspace ONE Access. (1967245)

When your Horizon Cloud environment is integrated with Workspace ONE Access, your end
users access their assigned desktops and applications using Workspace ONE Access. Due
to this know issue, the users see the application names displayed with lowercase characters,
regardless of the actual case used in the application names. This limitation is due to the way
Workspace ONE Access creates launch IDs from Horizon Cloud by using older Horizon Cloud
REST APIs. Workaround: None.

The memory usage percentages reported for desktop health reports and used for the desktop
health alerts are based on percentage of committed memory, which equals physical memory
plus pagefile size, and not on percentage of only physical memory. (2015772)

Committed memory for a desktop VM is calculated as physical memory plus pagefile size.
When calculating the percentage of memory usage in a desktop, the system takes the
percentage used of that total (physical memory plus pagefile size). Both the desktop health
alerts and the memory usage report in the desktop health reports use that percentage
calculation. However, when you log into a desktop VM and open the Windows Task Manager
to view the memory usage in the desktop's Windows operating system, the Windows Task
Manager displays percentage based on physical memory only. As a result, the memory usage
percentage that the desktop's Windows Task Manager displays does not match the memory
usage percentage displayed in the Desktop Health reports or in the desktop health alert.
Workaround: Keep in mind this difference if you decide to make a comparison between the
memory usage percentage reported by a desktop’s Windows Task Manager and the memory
usage percentage reported in the console's Desktop Health report and desktop health alerts for that desktop.

**If a desktop VM's CPU usage is at or close to 100%, the desktop alert is not triggered.** (1446496)

If an application or something in the desktop VM causes the VM's CPU usage to reach 100%, the desktop agent fails to send as many data samples as it usually sends to Horizon Cloud because the CPU is very busy. As a result of the low sample count returned, the calculation the system uses to trigger the desktop alert is affected. Workaround: None.

**End User, Horizon Agent, Horizon Client Related Known Issues**

The known issues listed here apply to pods deployed in Microsoft Azure.

**For a VM running Microsoft Windows 10 Enterprise multi-session 2004 or later, the DPI Synchronization and Display Scaling features have issues (2587685, DPM-6352)**

Due to an inability to query the current DPI in VMs running Microsoft Windows 10 Enterprise multi-session 2004 or later, these features with those VMs do not work as documented in the Horizon Client documentation. The DPI Synchronization and Display Scaling features do not work for PCoIP session reconnections. The DPI Scaling feature does not work for Blast session reconnections. Workaround: Log out of the session and log back in again.

**For a VM running Microsoft Windows 10 Enterprise operating system 1903 or later, the DPI Synchronization and Display Scaling features have issues (2589129)**

Due to an inability to query the current DPI in VMs running Microsoft Windows 10 Enterprise client operating system 1903 or later, when reconnecting a PCoIP or Blast session, the features do not work as documented in the Horizon Client documentation. Workaround: Log out of the session and log back in again.

**Sometimes when launching a VDI desktop using VMware HTML Access, an error message about being disconnected appears, and then subsequently the launch is successful. (2243471)**

VDI desktop virtual machines have a default session connection timeout, and when that timeout is reached, the session is disconnected. Sometimes, when launching a desktop, if the end user's HTML Access session has timed out at the time the desktop's default session connection timeout is reached, the desktop will initially throw that error, and then continue launching the desktop. Workaround: None.

**When a VDI desktop assignment has disk encryption selected and a one- or two-core VM model, and a desktop's underlying VM is powered off, the Horizon Client's automatic retry option might fail to make a connection. (2167432)**

When a VDI desktop VM is powered off due to the VDI desktop assignment’s power management settings, the VM has to power on and get ready before an end user connection can be made to that desktop. When an end user’s client tries to connect to a VDI desktop assignment’s VM and the VM is powered off, the system starts powering on that VM. For non-encrypted VMs, the VM is typically ready to accept a client connection in under 10 minutes.
However, an encrypted VM with one or two cores usually takes longer than 10 minutes to get ready to accept a connection. The Horizon Client's **Client Retry** option has an upper limit of 12 minutes. Because of this upper limit of the **Client Retry** option, when the end user has the client automatically retry the connection while the desktop's underlying VM is getting powered on and ready but the connection is not made within 12 minutes, the client's automatic retry gives up. Because an encrypted VM usually takes longer than 12 minutes until it is ready to take the client connection, the end user might see that Horizon Client's automatic retry fails to complete the connection to their encrypted desktop VM. Workaround: When you want to have disk encryption for a VDI desktop assignment, select a VM model that has more than two cores. Otherwise, if your VDI desktop assignment has disk encryption and has a VM model with one or two cores, inform your end users that they might experience this issue with using the **Client Retry** option with these encrypted desktop VMs.

**For a virtual desktop from a dedicated VDI desktop assignment, the shortcut link on the Horizon client’s Recent page might not launch the desktop.** (1813881, HD-3686, DPM-1140)

The iOS and Android versions of the Horizon clients have a Recent page which displays links to recently launched desktops. When the user does the initial launch of a dedicated pool virtual desktop, the desktop launches as usual, and the client creates a launch icon on the Recent page. However, when the user disconnects from the desktop and then tries later to launch the desktop from the Recent page, the desktop fails to launch because the launch icon is using a shortened version of the desktop name. Workaround: Launch the desktop from the client’s main page, and not the Recent page.

**Pods at 1976.0 manifest version and farm VMs running agent level 19.4: Users get disconnected after one hour from their desktops or remote application sessions when using HTML Access (Blast) and PCoIP protocols.** (2519400)

This issue is due to an issue in Microsoft terminal services in Microsoft Windows 10 Enterprise multi-session systems. For session-based desktops and remote applications provisioned from RDSH farms based on the Microsoft Windows 10 Enterprise multi-session operating system, when an end user reconnects to an existing desktop or remote application session using either HTML Access (Blast) or PCoIP protocol, after an hour has passed, the user's session is forcefully disconnected. There is no data loss. Even though the user can reconnect again and the session is in the same state it was at the disconnect time, this behavior repeats and the reconnected session is again forcefully disconnected after an hour.

This issue is resolved using Horizon Agents Installer (HAI) 20.1 or later. When your 1976.0 pod is updated to manifest 1976.1 or later, the Import Virtual Machine from Marketplace wizard will automatically install the agent software that has this fix. If your pod is still at 1976.0 manifest level, running the wizard will still install the agent software with the issue. However, when you seal the VM, the Images page will show the blue dot, signifying that you can use the Update Agent feature to update the agent to the level with the fix.

**Pods at manifest versions earlier than 2298: When switching protocols in the client, if you choose the Connect choice instead of Log Out and Reconnect, the client might become unresponsive.** (2528014)
This issue is resolved in pods that are updated to manifest 2298 or later. This issue happens when switching protocols in the client after establishing a session to an RDSH farm using one protocol. When you launch the desktop or application using one protocol, disconnect that session, use the client's menu to switch to a different protocol, launch the same desktop or application, the client presents a dialog box saying "This desktop is open on the server but is running a different protocol." and presents a choice to connect or to log out and reconnect. If you select the Connect button, the dialog appears a second time, and if you select Connect again, the client becomes unresponsive.

When using the Update Agent feature to update images that have an agent version earlier than 18.2.2, the update process might fail (2200962)

Images that you created on nodes at manifest level earlier than 965 might experience this issue. Sometimes the image has RunOnce registry values that block completion of the agent update process. Workaround: Perform the agent update again, adding the following command line argument on the Command Line tab of the Agent Update wizard: VDM_SUPPRESS_RUNONCE_CHECK=1
Basic Horizon Tenant Concepts - the Cloud Service, Control Plane, Horizon Universal Console, and Cloud-Connected Pods

This article gives a brief overview of some basic concepts involved in using the Horizon Control Plane and Horizon Cloud Service. The overall tenant environment consists of the VMware cloud-based service, the control plane, and your pods deployed into their on-premises, VMware SDDCs, or public cloud environments and connected to the control plane. From the single cloud-based Horizon Universal Console, you can efficiently deploy, manage, and monitor virtual desktops and apps across your pod fleet, regardless of where the pods are physically located.

Horizon Control Plane

VMware hosts the control plane in the cloud. Each control plane service works to simplify management of your Horizon environments and your virtual desktops and apps. When you sign up for a Horizon subscription license, VMware creates your tenant environment in this control plane and configures it according to the terms in that license.
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.

For an in-depth look at the control plane, see the Tech Zone’s Horizon Control Plane Services Architecture.

### Horizon Universal Console

The control plane also hosts the common cloud- and web-based management user interface called the Horizon Universal 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 dynamically reflects your tenant’s current state and is accessible from anywhere at any time, providing maximum flexibility.

The following screenshot is an illustration of the console’s Capacity page’s Pods tab for a tenant with four pods in its pod fleet.

![Horizon Universal Console](image)

#### Cloud-Connected Pods

In Horizon deployments, a pod is primarily a conceptual entity. A pod is based on various software components deployed into a supported environment, such as a public cloud, a VMware SDDC, or an on-premises data center. The pod’s inter-related components provide for the provisioning of virtual desktops and apps and for facilitating the routing of an end-user client’s request to a virtual desktop or app entitled for their use.
The specific collection of software components that make up a cloud-connected pod will depend on the type of deployment used to construct the pod. The current service release supports use of the following pod constructions with your tenant.

**Horizon pod**

Built on the Horizon Connection Server software and related software components. The components are deployed according to an architecture that VMware supports for use with such pods, such as on-premises, all-in-SDDC architecture, or federated architecture. These deployments involve a VMware SDDC in some way or form. This pod construction is named a Horizon pod because its underlying software is the Horizon Connection Server. For a brief overview, see [Horizon Pod Deployment Architectures](#).

**Horizon Cloud pod**

Built on the Horizon Cloud pod-manager technology for use with Microsoft Azure cloud and Microsoft Azure Virtual Desktop. The pod components are deployed by running the Horizon Cloud on Microsoft Azure deployment wizard. The wizard automates the pod deployment into your Microsoft Azure subscription. For an overview, see [Horizon Cloud on Microsoft Azure](#).

**An Initial Tenant Environment**

A tenant environment starts fresh, a clean-slate, without any cloud-connected pods. The first required step is to onboard a pod into that clean-slate environment. That pod will become the tenant’s very first cloud-connected pod.

The screenshot that follows is an illustration of what a brand new fresh Horizon Cloud environment looks like when an administrator logs in for the first time. This clean-slate screen is oriented around the pod types that can onboard to the service: Horizon pods and Horizon Cloud pods in Microsoft Azure. Until you have one pod onboarded and the Active Directory domain registration completed, all of the other UI pages located in the left-hand navigation are inaccessible.
To learn about the items needed to add the first pod to your pod fleet, see the Requirements Checklist that corresponds to the pod's type - Horizon pod or Horizon Cloud pod.

- Horizon pod - requirements checklist
- Horizon Cloud pod - requirements checklist
The purpose of this checklist is to inform you about the required elements for a Horizon Cloud on Microsoft Azure deployment. Following this checklist provides for successfully running the pod deployer and completing the day-1 tasks.

New

These checklist areas are updated for this release:

- Microsoft Azure Capacity Requirements - The transient pod deployment engine, also known as the jump box, is removed from the architecture for new deployments and updates. Capacity for a transient jump box would be required only if you open a support request and VMware Support determines the way to service that request is to deploy a support-related jump box VM, under their supervision.

- Horizon Cloud Golden Images, Desktops, and Farms:
  - A change of Azure VM family series that the Import Virtual Machine from Marketplace wizard uses.
    - For GPU-enabled VMs, the system now uses the Standard_NV12s_v3 model by default, for both single- and multi-pod images (Previously for GPU-enabled VMs the system used Standard_NV6 for Windows 10 and Windows Server operating systems and Standard_NC6s_v3 for Windows 11 operating systems.)
    - For non-GPU-enabled VMs, non-Windows 11, the system now uses the Standard_DS2_v2 for all of the non-Windows 11 operating systems. (Previously for single-pod images with Windows Server operating systems, the import workflow from the Imported VMs page used Standard_D2_v3.)
For non-GPU-enabled VMs, Windows 11, the system continues to use Standard_D4s_v3.

Checklist Audience

This checklist is primarily for Horizon Cloud customer accounts that have never had a Horizon Cloud on Microsoft Azure deployment in their tenant environment prior to the v2207 service release date. You might hear such tenants referred to as clean-slate environments or greenfield environments.

The set described here is primarily for production deployments. Trials and most proof-of-concept deployments can usually be handled with a subset, such as illustrated in Get Started with a Simplified, Proof-of-Concept Horizon Cloud Service on Microsoft Azure Pod Deployment.

Some of the sections listed below must be put into place prior to running the New Pod deployment wizard.

Some items can be deferred until after the deployment is completed and running.

Some items are listed as optional, because they relate to choices you make for your Horizon Cloud on Microsoft Azure deployment.

For example, you can run the New Pod wizard without choosing a Unified Access Gateway configuration and add the gateway configuration later. In this case, you wouldn’t need to fulfill the Unified Access Gateway requirements until that later time.

<table>
<thead>
<tr>
<th>Fulfill these before running the New Pod wizard</th>
<th>Can be done after the pod is deployed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Control plane tenant account</td>
<td>Active Directory</td>
</tr>
<tr>
<td>Microsoft Azure subscription items</td>
<td>Optional Unified Access Gateway (added post-deployment)</td>
</tr>
<tr>
<td>Networking</td>
<td>Universal Broker</td>
</tr>
<tr>
<td>Ports and protocols</td>
<td>DNS records</td>
</tr>
<tr>
<td>Optional Unified Access Gateway items</td>
<td>Golden images, desktop, farms capacities</td>
</tr>
<tr>
<td></td>
<td>Microsoft Windows operating system licenses</td>
</tr>
</tbody>
</table>

Some Key Considerations

When you are doing a trial or proof-of-concept Horizon Cloud on Microsoft Azure deployment, you might be the owner of the Microsoft Azure subscription that you will be using for the deployment or you might be doing the proof-of-concept on behalf of an organization that owns the subscription.

The owner of the subscription must ensure that none of the Microsoft Azure Policies that are in effect in the pod's subscription would block, deny, or restrict creation of the pod's components.
The reason for ensuring that is because during the pod deployment process, the pod deployer uses API calls to create resources within the subscription specified in the New Pod wizard. If that subscription has Microsoft Azure Policies in effect that would block, deny, or restrict creation of the pod's components, the deployment will be unsuccessful and require a support request to VMware support.

One example is the pod deployer requires that none of the Microsoft Azure Policies that are in effect in the pod's subscription are blocking, denying, or restricting creation of components on Azure storage account.

Before running the New Pod wizard, verify with the subscription's owner that the Microsoft Azure Policies Built-in Policy definitions do not block, deny, or restrict creation of the pod's components.

### Horizon Cloud Control Plane Requirements

| ☐ | VMware Customer Connect account that has been configured by VMware to log in to the Horizon Cloud control plane. For an overview of the relationship of this account to your Horizon Cloud tenant account, you can refer to the page Horizon Service Deployments and Onboarding Pods. |

### 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 obtain 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 you to use the Microsoft Azure portal and perform the Preparing to Deploy a Horizon Cloud Pod on Microsoft Azure. |
| ☐ | Horizon Cloud app registration and client secret key created in the pod's subscription. See Create a Horizon Cloud App Registration in the Pod's Subscription. When you or your organization will be using the feature to deploy the external gateway configuration in a subscription separate from the pod’s subscription, that gateway subscription also needs a Horizon Cloud app registration and a client secret key. |
Creating the Horizon Cloud app registration in a subscription automatically creates a service principal, by standard Microsoft Azure behavior.

To this service principal, assign a role that allows for Horizon Cloud to make API calls in the pod’s subscription.

Usually the Contributor role is assigned at the pod’s subscription level.

Alternatively, the role can be custom role assigned at the pod’s subscription level.

When you are deploying the external gateway configuration in an existing resource group in a separate subscription, you can assign either a custom role or the Contributor role for that gateway subscription’s service principal.

When you or your organization want the Horizon Cloud app registration to use a custom role, see the following page that describes the actions that custom role must have — When Your Organization Prefers to Use a Custom Role for the Horizon Cloud App Registration.

**Note** The role must be assigned directly to the Horizon Cloud app registration’s service principal. The use of a group-based assignment of a role to this service principal is unsupported.

Required resource providers registered in each Microsoft Azure subscription. See Register Resource Providers.

Subscription ID, directory ID, application ID and key identified for the subscription that you will specify in the deployment wizard.

The subscription must allow use of the Azure StorageV2 account type. Ensure that the subscription’s Microsoft Azure Policies do not restrict or deny creation of content that requires the Azure StorageV2 account type.

The subscription must allow creation of resource groups that do not have tags on them, unless you specify custom resource tags in the pod deployment wizard. The pod deployment process creates resource groups in the pod’s subscription without tags on them unless you specify custom resource tags in the wizard.

If you are not planning to use the deployment wizard’s custom resource tags feature, you must verify that your Microsoft Azure Policies allows creation of the pod’s untagged resource groups in the target subscription. If you do not specify custom resource tags in the wizard and your Microsoft Azure subscription has any type of resource tag requirement, pod deployment will fail when you try to deploy a pod into that subscription, or it will fail at the time of pod updates or adding a gateway configuration to a pod. For the names of the deployer-created resource groups, refer to Resource Groups that the Pod Deployer Creates.

Optional. Your organization might specify that they require you use a specific, pre-created resource group named by the organization for the external Unified Access Gateway, in a separate VNet and subscription from the pod’s subscription, and your organization requires use of a specific, pre-created resource group named by the organization, you would use the feature to deploy the external Unified Access Gateway into your own named resource group. Without using that feature, the pod deployer automatically creates a resource group with its own naming convention.

Use of the feature to use a requires you to create that resource group in that subscription before you run the pod deployer. You also must 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 to include in the custom role, see When Your Organization Prefers to Use a Custom Role for the Horizon Cloud App Registration.

Microsoft Azure Capacity Requirements

Where the following table refers to Microsoft Azure capacity, no manual installation is required. As long as the stated capacities are available in the subscription, the pod deployer will automatically instantiate the described VMs.
For capacity involving VM families, the Microsoft Azure portal also uses the term *quota*.

<table>
<thead>
<tr>
<th>☐</th>
<th>Microsoft Azure capacity for the core Horizon Cloud pod artifacts to deploy into that subscription. (This list excludes the capacities needed for the optional Unified Access Gateway configurations and for your anticipated desktop and app workloads.)</th>
</tr>
</thead>
</table>
| **Pod** | Pod Managers — 2 x Standard_D4_v3 (if no Standard_D4_v3 in the region, 2 x Standard_D3_v2)  
Microsoft Azure Database for PostgreSQL Service — Generation 5, Memory Optimized, 2 vCores, 10 GB Storage  
When your pod is ready to use, your capacity in Microsoft Azure cloud will have to also accommodate the imported VMs, golden images, virtual desktops, RDSH farms, and App Volumes app-capture VMs that you create in that pod. See the Horizon Cloud Golden Images, Desktops, and Farms section below.  
When your tenant account is enabled to use the Horizon Infrastructure Monitoring feature and you are planning to enable that feature on the pod after the pod is deployed, your capacity will have to also accommodate at that time the deployment of the Horizon Edge Virtual Appliance. See the Horizon Infrastructure Monitoring Requirements section below.  
In a special circumstance where you open a support request and VMware Support determines the way to service that request is to temporarily deploy a support-related jump box VM, your capacity will have to accommodate the deployment of a Standard_F2 model VM at that time. |
| ☐ | Optional. Capacity required when specifying use of Unified Access Gateway for the pod.  
**Note** The A4_v2 VM model is only sufficient for proofs-of-concept (PoCs), pilots, or smaller environments where you know that you will not exceed 1,000 active sessions on the pod. |

**External Unified Access Gateway in the pod’s same VNet**

- 2 x Standard_A4_v2 or 2 x Standard_F8s_v2

**External Unified Access Gateway in its own VNet**

- External Gateway Connector — 1 x Standard_A1_v2  
- External Unified Access Gateway — 2 x Standard_A4_v2 or 2 x Standard_F8s_v2.

**Internal Unified Access Gateway**

- 2 x Standard_A4_v2 or 2 x Standard_F8s_v2

If your subscription’s region does not provide for the Standard_F8s_v2 VM sizes, the pod deployer wizard will not display that choice in the selector in the Specify Gateways wizard step.
Network Requirements

- Microsoft Azure Virtual Network (VNet) created in your target Microsoft Azure region with applicable address space to cover required subnets. See Configure the Required Virtual Network in Microsoft Azure. 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.

  **Tip** After the pod is deployed, you can edit the pod to add 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 Overview of Using Multiple Tenant Subnets.

- 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 and Related Service Features and Horizon Cloud Pod - Ports and Protocols Requirements.

- Optional. 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, when you want networking between the VNet and your on-premises corporate network.

Ports and Protocols Requirements

- Specific ports and protocols are required for onboarding pods and ongoing operations of your Horizon Cloud environment. See Horizon Cloud Pod - Ports and Protocols Requirements.
Unified Access Gateway Requirements

You can run the New Pod wizard without choosing a Unified Access Gateway configuration and add the gateway configuration later. In this case, you wouldn't need to fulfill the Unified Access Gateway requirements until that later time. By definition, an external Unified Access Gateway enables clients on external networks to launch the virtual desktops and apps, while an internal Unified Access Gateway enables clients on internal networks to have trusted HTML Access (Blast) connections. To support end-user connections from the Internet and launching their virtual desktops and apps, the pod must have an external Unified Access Gateway configured.

If you do select the Unified Access Gateway options within the New Pod wizard, the wizard mandates specific items below.

| ☐ | FQDN for the Unified Access Gateway configuration. |
| ☐ | Certificate or certificates for the Unified Access Gateway in PEM format matching the FQDN. |

**Note** 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.

| ☐ | Optional, except when you want your end users to use two-factor authentication. In that case, the Unified Access Gateway must be configured for two-factor authentication with one of the authentication system types supported for use with Horizon Cloud on Microsoft Azure deployments. The configuration should include: |

- DNS Addresses for Unified Access Gateway to resolve the name of that authentication server
- Routes for Unified Access Gateway to resolve network routing to that authentication server

**Note** After the pod is deployed and you have configured two-factor authentication in the Universal Broker settings, some additional configuration is needed when you want to also have your internal end users skip using that two-factor authentication. When a pod has an internal Unified Access Gateway configuration, that configuration routes the connection requests to virtual desktops and apps for such internal end users. When you want Universal Broker to skip the two-factor authentication step for your internal end users, then after the pod is deployed and Universal Broker is configured, enter the ranges of egress NAT addresses that correspond to your internal end-user traffic. Those ranges let Universal Broker identify client traffic from the internal end users distinct from the external end users for the purposes of skipping the two-factor authentication. For details, see the documentation topic Define Internal Network Ranges for Universal Broker.

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 Horizon Cloud on Microsoft Azure - First Pod Deployment - High-Level Workflow 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 console’s Active Directory registration workflow mandates the following items. For full 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 Horizon pods that are cloud-connected to the same customer account using the Horizon Cloud Connector.

☐ Domain Bind Account

Active Directory domain bind account (a standard user with read access) that has the sAMAccountName attribute. The sAMAccountName attribute must be 20 characters or less and cannot contain any of the following characters: */ \ ( ) : ; | = , + * ? < >

The account must have the following permissions:
  - List Contents
  - Read All Properties
  - Read Permissions
  - Read tokenGroupsGlobalAndUniversal (implied by Read All Properties)

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

  - 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.

  - 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.

Reference: Service Accounts That Horizon Cloud Requires for Its Operations
Auxiliary Domain Bind Account

Must be separate from the main domain bind account. The UI will prevent re-using the same account in both fields.

Active Directory domain bind account (a standard user with read access) that has the sAMAccountName attribute. The sAMAccountName attribute must be 20 characters or less and cannot contain any of the following characters: */ \ { } : ; | = , * ? < >

The account must have the following permissions:
- List Contents
- Read All Properties
- Read Permissions
- Read tokenGroupsGlobalAndUniversal (implied by Read All Properties)

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

- 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.
- 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.

Reference: 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 the virtual computers to the domain. Typically a new account that you create for this express purpose. (A domain join user account)

The account must have the sAMAccountName attribute. The sAMAccountName attribute must be 20 characters or less and cannot contain any of the following characters: */ \ [ ] : ; | = , + * ? < >

The use of white spaces in the account's user name is currently unsupported.

You should also set the account password to Never Expire to ensure continued ability for Horizon Cloud to perform the Sysprep operations and join the virtual computers to the domain.

This account requires the following Active Directory permissions, applied to the Computers OU, or to the OU that you will enter into the console’s Domain Join UI.

- Read All Properties - this object only
- Create Computer Objects - this object and all descendant objects
- Delete Computer Objects - this object and all descendant objects
- Write All Properties - Descendant Computer objects
- Reset Password - Descendant Computer objects

Regarding the target Organizational Unit (OU) that you plan to use for farms and VDI desktop assignments, this account also requires the Active Directory permission named Write All Properties on all descendant objects of that target Organizational Unit (OU).

For more details, see Service Accounts That Horizon Cloud Requires for Its Operations

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.
Optional Auxiliary Domain Join Account

Active Directory domain join account which can be used by the system to perform Sysprep operations and join the virtual computers to the domain. Typically a new account that you create for this express purpose. (A domain join user account)

The account must have the `sAMAccountName` attribute. The `sAMAccountName` attribute must be 20 characters or less and cannot contain any of the following characters: */ \ [ ] : ; | = , + * ? < >

The use of white spaces in the account's user name is currently unsupported.

You should also set the account password to Never Expire to ensure continued ability for Horizon Cloud to perform the Sysprep operations and join the virtual computers to the domain.

This account requires the following Active Directory permissions, applied to the Computers OU, or to the OU that you will enter into the console’s Domain Join UI.
- Read All Properties - this object only
- Create Computer Objects - this object and all descendant objects
- Delete Computer Objects - this object and all descendant objects
- Write All Properties - Descendant Computer objects
- Reset Password - Descendant Computer objects

Regarding the target Organizational Unit (OU) that you plan to use for farms and VDI desktop assignments, this account also requires the Active Directory permission named Write All Properties on all descendant objects of that target Organizational Unit (OU).

For more details, see Service Accounts That Horizon Cloud Requires for Its Operations

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. This group is granted the Super Administrator 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 Cloud supports use of Active Directory security groups for both administrator logins and end user entitlements. Nested groups are supported. Group membership is evaluated by requesting the `tokenGroups` computed attribute, meaning that Horizon Cloud has no nesting depth limitations but supports whatever is set in your Active Directory.

When asked if there are additional considerations or additional limitations in the context of Active Directory groups and the combination of Horizon Cloud plus Universal Broker plus Workspace ONE Access Cloud, the answer to that question is no, none.

If your tenant environment has any Horizon Cloud pods in Microsoft Azure running manifests older than manifest 1600.0, the domain join account and any auxiliary domain join accounts must also be in Horizon Cloud Administrators group — or in an Active Directory group which is granted the Super Administrator role in Horizon Cloud.

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

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.

If you want to use an Active Directory configured for LDAPS, you must request enablement of the LDAPS supported features with your Horizon Cloud tenant. For more details, see Using Active Directory Environments Configured for LDAPS.

Universal Broker Configuration

For the Universal Broker configuration, ensure you fulfill the items from the following table that are applicable to your desired options. For full details, see Configure Universal Broker.

Outbound Internet access on the V Nets 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 and Related Service Features and Horizon Cloud Pod - Ports and Protocols Requirements.

Depending on the types of end-user connections you want to provide for:

- For end-user connections from the Internet and launching their virtual desktops and apps, a pod must have an external Unified Access Gateway configured.
- If all of your end-user connections will be always from your internal network, no Unified Access Gateway is required on the pod — except when you want Universal Broker to enforce two-factor authentication with those internal end-user connections.
Optional. When you want Universal Broker to enforce two-factor authentication, pods must have an external Unified Access Gateway configured for two-factor authentication with one of the authentication system types supported for use with Horizon Cloud on Microsoft Azure deployments. Universal Broker passes the authentication request to Unified Access Gateway, which communicates with the authentication server, and then relays the response back to Universal Broker. That external Unified Access Gateway configuration requires the following items:

- 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. If you want to use the VMware-provided brokering FQDN, a custom FQDN is not required.

**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. For background information about DNS record mapping, see the Microsoft cloud services documentation page Configuring a custom domain name for an Azure cloud service.

**Note** If you deployed the pod with the external and internal gateway configurations having the same FQDN, then after pod deployment, 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.

**When you configure the tenant's 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.

**When the pod has an external Unified Access Gateway**

Create a public DNS record for external end-user access that matches the FQDN on the external gateway configuration. This DNS record points that FQDN to the Microsoft Azure external load balancer in the pod’s external Unified Access Gateway configuration. For background information about DNS record mapping, see the Microsoft documentation page Configuring a custom domain name for an Azure cloud service.

**When the pod has an internal Unified Access Gateway**

Create an internal DNS record for internal end-user access that matches the FQDN on the internal gateway configuration. This DNS record points that FQDN to the Microsoft Azure internal load balancer in the pod’s internal Unified Access Gateway configuration.
# 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 or more of the supported Microsoft Azure VM configurations. |
|   | - Standard_DS2_v2 |
|   | - Standard_NV12s_v3 (for the service’s automated Import VM from Marketplace wizard or manual import, and NVIDIA GRID graphics driver), Standard_NV8as_v4 (for the manual import method and AMD graphics driver) |
|   | - Standard_D4s_v3 |

When using the console’s 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 its internal settings and the specific operating system (OS).

The system uses the models as indicated for both single-pod images and multi-pod images.

**Import VM from Marketplace wizard creates:**

- Non-GPU, non-Windows 11, a Standard_DS2_v2 VM
- Non-GPU using Windows 11, a Standard_D4s_v3 VM
- GPU-capable, a Standard_NV12s_v3 VM

| ☐ | 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 configurations 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 configurations not compatible with Horizon Cloud RDS farm operations. |

This requirement also applies to a VM running Microsoft Windows 10 Enterprise multi-session or Windows 11 Enterprise multi-session when that VM is used with Horizon Cloud. As described in the Microsoft Windows Enterprise multi-session FAQ in the Microsoft Azure Virtual Desktop documentation, Microsoft Windows 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 Enterprise multi-session.

For production environments, VMware scale testing recommends using models that have a minimum of 2 CPUs or larger.
Image Management Service (IMS) Requirements

Starting with the July 2021 release, when your Horizon Cloud pods are all running manifest 2632 or later and your tenant has Universal Broker enabled, the Horizon Image Management Service features are available to use with those pods. Use of the multi-pod image features provided by the service has additional requirements. For full details of the system requirements for use of those features, see the Microsoft Azure section of Horizon Image Management Service System Requirements. An outline of the additional requirements on the pod’s subscription and that subscription’s Horizon Cloud app registration and its service principal when using multi-pod images are outlined in the following table.

| ☐ | The subscriptions used by those pods that are participating in multi-pod images must be within a single Microsoft Azure Active Directory (AAD) tenant. |
| ☐ | The Horizon Cloud app registration’s service principal used by those pods that are participating in multi-pod images must meet one of the following requirements:  
  - Same service principal used across all those participating pods and subscriptions.  
  - Each service principal must have read access to every Microsoft Azure subscription used by those participating pods.  
Because the pods are likely to be in different subscriptions, the read-access requirement enables each participating pod’s subscription to have line of sight to the other participating pods’ subscriptions. This line of sight is needed for the service to use the features of Azure Shared Image Gallery for creating the multi-pod images. |
| ☐ | Any custom roles that are used by the participating pods’ service principals must include the following permissions related to use of Azure Shared Image Gallery.  
  - Microsoft.Compute/galleries/read  
  - Microsoft.Compute/galleries/write  
  - Microsoft.Compute/galleries/delete  
  - Microsoft.Compute/galleries/images/*  
  - Microsoft.Compute/galleries/images/versions/* |

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 Azure Virtual Desktop licensing specific to Windows 11 Enterprise multi-session, Windows 10 Enterprise multi-session and Windows 7 Enterprise, see the Microsoft Azure documentation topic [Azure Virtual Desktop pricing](#).

| ☐ | Licensing for one or more of the following types: Microsoft Windows 7 Enterprise, Microsoft Windows 10, Microsoft Windows 11 (single-session, VDI client types) |
| ☐ | Licensing for Microsoft Windows 10 Enterprise multi-session or Microsoft Windows 11 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 |

### Horizon Infrastructure Monitoring Requirements

If your Horizon Cloud tenant is enabled to use Horizon Infrastructure Monitoring, you can use the Horizon Universal Console to activate that feature on your choice of pods. When you choose to activate that feature on a pod, the system automatically builds a virtual appliance into the same Microsoft Azure subscription where that pod resides, and configures that appliance to collect the infrastructure data. Before you can activate this monitoring feature, your tenant must be onboarded to VMware Cloud Services. See [Onboard Your Horizon Cloud Tenant to VMware Cloud Services](#) and [Horizon Infrastructure Monitoring and the Pods in Your Horizon Cloud Environment](#).

| ☐ | Onboard your tenant to VMware Cloud Services. |
| ☐ | For each pod on which you will activate the Horizon Infrastructure Monitoring feature, the pod's Microsoft Azure subscription must accommodate these additional requirements. |
  |   ✓ Horizon Edge Virtual Appliance — 1 x Standard_D4_v3 |

### Reference Architecture

Use the architecture diagrams below for reference.
Figure 3-1. Illustration of the Horizon Cloud Pod Architecture where the Pod has 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 3-2. 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
VMware Horizon Pods with Horizon Cloud Control Plane - Requirements Checklist - Appropriately Updated Starting from the v2204 Service Release

Complete the following tasks to prepare your Horizon pod’s components for onboarding the pod to the Horizon Cloud control plane. Ensure that the requirements are satisfied as described in the following sections to complete a successful onboarding.

Checklist Audience

This checklist is primarily for clean-slate, greenfield, Horizon Cloud customer accounts that have never had a pod onboarded to their tenant environment prior to the v2204 service release date.

Some of the requirements listed in the following sections are the ones needed for successfully onboarding a Horizon pod for the purposes of using a subscription license with the pod. Some requirements are those needed for the key tasks that are performed after that initial onboarding to enable the use of Horizon Cloud Control Plane services with the pod.

For reference, the high-level workflow of cloud-connecting a Horizon pod is described in Onboarding a Horizon Pod to Horizon Cloud Control Plane.

Horizon Cloud Control Plane Requirements

- Active VMware Customer Connect account to log in to the Horizon Cloud control plane. (The name My VMware was the prior name for VMware Customer Connect.)

- Valid Horizon Universal License. For more information about this license, see the Horizon Universal License page.
## Horizon Pod and Horizon Cloud Connector Requirements

| ☐ | As the VMware Interoperability Matrix indicates, for interoperability between Horizon Cloud Connector versions and Horizon versions, the Horizon pod must be running no earlier a version than 7.13.0. Then, to use 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. |
| ☐ | As of this writing, for new deployments, it is strongly recommended to use Horizon Cloud Connector version 2.2.x or later. Versions earlier than the latest version 2.2.x should not be used for new deployments because they will not contain the latest fixes and improvements. To use the latest cloud services and features with the cloud-connected pod and have the latest security fixes, it must be running the most current Horizon Cloud Connector version. The Horizon Cloud Connector appliance deployment procedure uses:  
  - Static IP  
  - DNS forward and reverse lookup records |
Resource requirements for the Horizon Cloud Connector virtual appliance. The resource requirements depend on the deployed Horizon pod’s architecture: all-in-SDDC or federated architecture. The lists below reflect which versions are currently supported for new deployments for each design.

**All-in-SDDC Architecture**

In the all-in-SDDC architecture, the OVA format of the appliance is deployed within a VMware SDDC.

**Version 2.1.2**
- Primary node: 4 vCPUs, 8 GB memory (RAM), 40 GB datastore
- Each additional worker node: 4 vCPUs, 8 GB memory (RAM), 40 GB datastore

**Version 2.2.x**
- Primary node: 4 vCPUs, 8 GB memory (RAM), 40 GB datastore
- Each additional worker node: 4 vCPUs, 8 GB memory (RAM), 40 GB datastore

**Federated Architecture**

In the federated architecture, a cloud-native format of the appliance is deployed within the specific cloud-native infrastructure. These lists are the cloud-native formats currently supported and file downloads available. Which cloud-native formats are available will vary for each specific version of Horizon Cloud Connector.

The underlying machine instance you use to deploy the appliance must meet or exceed the following resource requirements.

**Version 2.1.2**
For these deployments, use of primary node only is supported.
- Azure VMware Solution (AVS) - 8 vCPUs, 32 GB memory (RAM), 40 GB datastore. The Azure Cloud instance Standard_D8_v3 is the support-validated instance that provides for the 8 vCPUs minimum.
- Google Cloud VMware Engine (GCVE) - 8 vCPUs, 32 GB memory (RAM), 40 GB datastore. The Google Cloud instance n2-standard-8 is the support-validated instance that provides for the 8 vCPUs minimum.
- VMware Cloud on AWS - 8 vCPUs, 16 GB memory (RAM), 40 GB datastore. The Amazon instance c5.2xlarge is the support-validated instance that provides for the 8 vCPUs minimum.

**Version 2.2.x**
For these deployments, use of primary node only is supported.
- Azure VMware Solution (AVS) - 8 vCPUs, 32 GB memory (RAM), 40 GB datastore. The Azure Cloud instance Standard_D8_v3 is the support-validated instance that provides for the 8 vCPUs minimum.
- Google Cloud VMware Engine (GCVE) - 8 vCPUs, 32 GB memory (RAM), 40 GB datastore. The Google Cloud instance n2-standard-8 is the support-validated instance that provides for the 8 vCPUs minimum.
- VMware Cloud on AWS - 8 vCPUs, 16 GB memory (RAM), 40 GB datastore. The Amazon instance c5.2xlarge is the support-validated instance that provides for the 8 vCPUs minimum.

Active Directory user mandated 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 VMware Horizon 7 or VMware Horizon 8 documentation's Horizon Administration guide or Horizon Console Administration guide that is applicable for your pod’s software version.
# 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 tenant must have line-of-sight to the same set of Active Directory domains at the time you onboard those pods to the cloud control plane. This line-of-sight requirement applies not only to additional Horizon pods that you subsequently onboard to your pod fleet after the first one, but also to Horizon Cloud pods deployed into Microsoft Azure using the same cloud tenant.**

- **Domain bind account.**
  - Active Directory domain bind account (a standard user with read access) that has the `sAMAccountName` attribute. The `sAMAccountName` attribute must be 20 characters or less and cannot contain any of the following characters: `"\[ ] : ; | = , + * ? < >`
  - The account must have the following permissions:
    - List Contents
    - Read All Properties
    - Read Permissions
    - Read tokenGroupsGlobalAndUniversal (implied by Read All Properties)
  - 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 sAMAccountName attribute. The sAMAccountName attribute must be 20 characters or less and cannot contain any of the following characters: "/\[\] ; : ; = , + * ? < >

- The account must have the following permissions:
  - List Contents
  - Read All Properties
  - Read Permissions
  - Read tokenGroupsGlobalAndUniversal (implied by Read All Properties)

- 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.
<table>
<thead>
<tr>
<th>Domain join account</th>
</tr>
</thead>
<tbody>
<tr>
<td>✔️ 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)</td>
</tr>
<tr>
<td>✔️ Set account password to Never Expire</td>
</tr>
<tr>
<td>✔️ This account requires the following Active Directory permissions: List Contents, Read All Properties, Read Permissions, Reset Password, Create Computer Objects, Delete Computer Objects.</td>
</tr>
<tr>
<td>✔️ 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 Universal Console.</td>
</tr>
<tr>
<td>✔️ For additional details and requirements, see Service Accounts That Horizon Cloud Requires for Its Operations</td>
</tr>
</tbody>
</table>

**Note**

- The use of white spaces in the domain join account’s logon name is not supported. If the domain join account's logon name contains a white space, unexpected results will occur in the system operations that rely on that account.

- 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.

- Prior to manifest 1600.0, this account required the permissions of the system’s Super Administrator role. If your tenant environment has any Horizon Cloud pods in Microsoft Azure running manifests older than manifest 1600.0, the domain join account must be in the described Horizon Cloud Administrators group. The system uses this domain join account with all of your tenant’s Horizon Cloud pods in Microsoft Azure. When your tenant has existing pods of manifests older than 1600.0, the domain join account must meet this requirement. For details, see Service Accounts That Horizon Cloud Requires for Its Operations.
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)
- 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 Universal Console.
- For additional details and requirements, see Service Accounts That Horizon Cloud Requires for Its Operations

**Note**
- The use of white spaces in the domain join account's logon name is not supported. If the domain join account's logon name contains a white space, unexpected results will occur in the system operations that rely on that account.
- 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.
- Prior to manifest 1600.0, this account required the permissions of the system's Super Administrator role. If your tenant environment has any Horizon Cloud pods in Microsoft Azure running manifests older than manifest 1600.0, the domain join account must be in the described Horizon Cloud Administrators group. The system uses this domain join account with all of your tenant's Horizon Cloud pods in Microsoft Azure. When your tenant has existing pods of manifests older than 1600.0, the domain join account must meet this requirement. For details, see Service Accounts That Horizon Cloud Requires for Its Operations.

**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 granted the Super Administrator 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.

**Note** If your tenant environment has any Horizon Cloud pods in Microsoft Azure running manifests older than manifest 1600.0, the domain join account and any auxiliary domain join accounts must also be in Horizon Cloud Administrators group — or in an Active Directory group which is granted the Super Administrator role in Horizon Cloud.

**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 with the Horizon Cloud Connector paired with that pod, and Horizon Cloud Connector with the Horizon Cloud control plane. See DNS, Ports, and Protocols Requirements When Using Horizon Cloud Connector and a Horizon Pod.
Universal Broker

When using the console to configure Universal Broker for your tenant, ensure you fulfill the items from the following table that are applicable to your desired options. For full details, see Configure Universal Broker.

| ☐ | Outbound communications from the pod’s paired Horizon Cloud Connector must resolve and reach specific DNS names using specific ports and protocols. This is required for Universal Broker configuration and ongoing operations. See Horizon Pods - Port and Protocol Requirements for Universal Broker. |
| ☐ | Depending on the types of end-user connections you want to provide for:  
  - For end-user connections from the Internet and launching their virtual desktops and applications, a pod must have an external Unified Access Gateway configured.  
  - If all of your end-user connections will be always from your internal network, no Unified Access Gateway is required on the pod — except when you want Universal Broker to enforce two-factor authentication with those internal end-user connections. |
| ☐ | Optional: A custom FQDN that your end users will use to access the Universal Broker service and the certificate based on that FQDN. If you want to use the VMware-provided brokering FQDN, a custom FQDN is not required. |
| ☐ | Optional. When you want Universal Broker to enforce two-factor authentication, pods must have an external Unified Access Gateway configured for two-factor authentication to an authentication server. Universal Broker passes the authentication request to Unified Access Gateway, which communicates with the authentication server, and then relays the response back to Universal Broker. That external Unified Access Gateway configuration requires the following items:  
  - 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 |

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 a Horizon Pod to Use Subscription Licenses or Control Plane Services with That Pod

You onboard 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. Such cloud-hosted 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 pod deployment and for enabling control plane services with that pod, such as the Cloud Monitoring Service (CMS) and Universal Broker.

For a high-level description of the process of first-time onboarding any of the supported pod types to the cloud control plane, see Horizon Service Deployments and Onboarding Pods — Horizon Pods and Horizon Cloud Pods on Microsoft Azure.
Important  Your tenant's pod fleet can consist of both Horizon Cloud pods and Horizon pods. All pods in the pod fleet must have line of sight to the same set of Active Directory domains. If your fleet already has Horizon Cloud pods and you are connecting your first Horizon pod, you must ensure that the Horizon pod will be able to have line of sight to the Active Directory domains that are already registered with your Horizon Cloud tenant at the time you connect that 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 supported public clouds such as 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**

After your Horizon pod is added to your cloud-connected pod fleet, you can leverage the cloud-hosted services, features, and workflows that Horizon Cloud provides, 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.

This chapter includes the following topics:

- **Horizon Pod Deployment Architectures**
- **Onboarding a Horizon Pod to Horizon Cloud Control Plane**
- **DNS, Ports, and Protocols Requirements When Using Horizon Cloud Connector and a Horizon Pod**
- **Horizon Pod and Horizon Cloud Connector - Preparing to Onboard to Control Plane Services**
- **Connect Horizon Cloud Service with an Existing Horizon Pod to Use Horizon Subscription Licenses or Cloud-Hosted Services or Both**

**Horizon Pod Deployment Architectures**

This article describes the different deployment architectures used by Horizon pods. Before onboarding a Horizon pod to Horizon Cloud, you must first identify its deployment architecture, which in turn determines the steps you must perform to deploy the Horizon Cloud Connector appliance into that architecture.
Horizon pods can be deployed using one of the following deployment architectures: **on-premises**, all-in-SDDC, or **federated**.

**On-premises Deployment Architecture**

When a Horizon pod is deployed on premises, the pod components are deployed as follows:

<table>
<thead>
<tr>
<th>These pod components:</th>
<th>Are deployed into:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Horizon Connection Server</td>
<td>The vSphere infrastructure of the on-premises VMware SDDC</td>
</tr>
<tr>
<td>Unified Access Gateway</td>
<td></td>
</tr>
<tr>
<td>App Volumes Manager</td>
<td></td>
</tr>
<tr>
<td>Virtual machines (VMs) for virtual desktops, published desktops, and published applications</td>
<td></td>
</tr>
</tbody>
</table>

**Note**  When onboarding an on-premises pod to Horizon Cloud, you must deploy Horizon Cloud Connector into the vSphere infrastructure of the on-premises VMware SDDC.
All-in-SDDC Deployment Architecture

When a Horizon pod is deployed into a cloud environment using all-in-SDDC architecture, the pod components are deployed as follows:

<table>
<thead>
<tr>
<th>These pod components:</th>
<th>Are deployed into:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Horizon Connection Server</td>
<td>The vSphere infrastructure of the VMware SDDC within the cloud environment</td>
</tr>
<tr>
<td>Unified Access Gateway</td>
<td></td>
</tr>
<tr>
<td>App Volumes Manager</td>
<td></td>
</tr>
<tr>
<td>VMs for virtual desktops, published desktops, and published applications</td>
<td></td>
</tr>
</tbody>
</table>

**Note** When onboarding an all-in-SDDC pod to Horizon Cloud, you must deploy Horizon Cloud Connector into the vSphere infrastructure of the VMware SDDC.
For example, consider an all-in-SDDC pod deployed into VMware Cloud on AWS. In this case:

- All pod components reside within the vSphere infrastructure of VMware Cloud on AWS.
- You must deploy Horizon Cloud Connector into the vSphere infrastructure of VMware Cloud on AWS

### Federated Deployment Architecture

When a Horizon pod is deployed into a cloud environment using federated architecture, the pod components are deployed as follows:

<table>
<thead>
<tr>
<th>These pod components:</th>
<th>Are deployed into:</th>
</tr>
</thead>
<tbody>
<tr>
<td>VMs for virtual desktops, published desktops, and published applications</td>
<td>The vSphere infrastructure of the VMware SDDC within the cloud environment</td>
</tr>
<tr>
<td>Horizon Connection Server</td>
<td>The native cloud infrastructure, outside the VMware SDDC</td>
</tr>
<tr>
<td>Unified Access Gateway</td>
<td></td>
</tr>
<tr>
<td>App Volumes Manager</td>
<td></td>
</tr>
</tbody>
</table>

**Note** When onboarding a federated pod to Horizon Cloud, you must deploy Horizon Cloud Connector into the native cloud infrastructure outside the VMware SDDC.
For example, consider a federated pod deployed into Google Cloud VMware Engine (GCVE). In this case:

- VMs for virtual desktops, published desktops, and published applications reside within the vSphere infrastructure of GCVE.
- Horizon Connection Server, Unified Access Gateway, and App Volumes Manager all reside in the native Google Cloud Platform (GCP) infrastructure, outside GCVE.
- You must deploy Horizon Cloud Connector into the native GCP infrastructure, outside GCVE.

**Onboarding a Horizon Pod to Horizon Cloud Control Plane**

This list is a high-level overview of the steps when you are onboarding your very first pod to the control plane and that pod is an existing Horizon pod that you have already stood up using one of the supported deployment architectures. A Horizon pod is based on the Horizon Connection Server software.

The following diagram illustrates the overall flow.

![Diagram](image)

**Reasons for Onboarding**

You onboard Horizon pods to the cloud for two primary use cases: to activate a subscription license for that pod and to activate your use of those cloud-hosted services that the cloud plane provides for Horizon pods, such as the Cloud Monitoring Service (CMS). The CMS is one of the central services provided in Horizon Cloud. The CMS provides for visibility, health monitoring, and help desk services with cloud-connected pods.

After you complete these onboarding steps for your very first cloud-connected pod, the subscription license takes effect for that onboarded pod. You can also start using the cloud-hosted services that the control plane provides for that pod type, which include the Cloud Monitoring Service (CMS). At that point, you can also onboard additional pods.
Have the Pod All Ready Before You Begin

Before beginning this workflow, you must have a Horizon pod deployed already, using one of the supported deployment designs. Obtain specific pod-deployment steps from the following resources.

- Horizon 7 Documentation, Horizon Documentation.
- Horizon on VMware Cloud on AWS product page.
- Tech Zone’s Horizon on Azure VMware Solution Architecture, Tech Zone’s Horizon on Azure VMware Solution Configuration.
- Tech Zone’s Horizon on Google Cloud VMware Engine Architecture.

Sequence

**Important**  Complete the full sequence to fully connect your first pod to Horizon Cloud before you start deploying the Horizon Cloud Connector with an additional Horizon pod. 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 fails. At that point, you must 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 include obtaining a Horizon subscription license, such as the Horizon universal license. For a high-level description of the process of onboarding a pod to the cloud control plane, also see Horizon Cloud Deployments and Onboarding Pods.

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 the 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 into your pod’s environment.
4 Optionally log in to the Horizon Cloud tenant portal and configure more 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 more administrators at the start of this process. If only a single My VMware account is associated with your tenant account and you lose your access to the credentials, delays might occur because you must 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 steps from Download and Deploy the Horizon Cloud Connector.

**Note** If you deploy Horizon Cloud Connector 1.9 and want to enable the use of Horizon Cloud services such as the CMS, you must select the *Full Feature* profile during deployment.

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. Follow the steps in Enable SSH Access to the Horizon Cloud Connector Using a Command Line Interface.

**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 activate and deactivate 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 1.6 or Later 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 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 with Horizon Control Plane 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 4 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. For a list of such typical tasks and links to steps, see Typical Administrative and Maintenance Tasks for a Paired Horizon Cloud Connector.

12 Register your Active Directory domain with the deployed pod, which includes providing the names of service accounts. Ensure these service accounts meet the requirements described in Service Accounts That Horizon Cloud Requires for its Operations.

**Tip** Completing the Active Directory domain registration workflow enables you to take advantage of all 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 areas of the console involving those features are inaccessible.

13 In the rare, atypical situation where your tenant environment already has Horizon Cloud pods in Microsoft Azure that are running manifests older than manifest 1600.0, you must give the Horizon Cloud Super Administrators role to an Active Directory group that includes the domain join account as a member. See the topic Assign Horizon Cloud Administrative Roles to Active Directory Groups in the 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.

**Note** (Horizon Cloud Connector 2.0 and later) Unless otherwise specified, the following DNS, ports, and protocols requirements apply alike to the primary node and worker node of the Horizon Cloud Connector appliance.

As described in Tight Integration Within the VMware Ecosystem, you can use Horizon Cloud with other products available from the broader VMware ecosystem. Those other products might have additional DNS requirements. Such additional DNS requirements are not detailed here. For such DNS requirements, see the documentation set for the specific products that you will be integrating with your cloud-connected Horizon pod.

DNS Requirements for Pod Connectivity and Service Operations that Apply on a Tenant-Wide Basis

This section describes the DNS requirements for pod connectivity and service operations that apply on a tenant-wide basis.

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 Domain Name Service (DNS) names are resolvable and reachable using the specific ports and protocols as listed according to the following tables.

**Important** Keep in mind the following important points:

- For all tenant accounts, reachability to DNS name `cloud.horizon.vmware.com` is required. Reachability to `cloud.horizon.vmware.com` is required in addition to reachability to the regional control plane DNS name for the region specified in your tenant account.

- Horizon Cloud Connector uses SSL certificates signed by DigiCert, an industry-trusted certificate authority (CA). These certificates use CRL (Certificate Revocation Lists) and OCSP (Online Certificate Status Protocol) queries that refer to specific DNS names on the DigiCert domain. To ensure Horizon Cloud Connector connectivity, you must configure these DNS names to be resolvable and reachable by the virtual appliance. If these DNS names are not reachable, you will not be able to access the Horizon Cloud Connector configuration portal. The specific names are determined by DigiCert, and therefore are not in VMware's control.

- If you plan to enable Universal Broker for use with the pod, there are connectivity requirements in addition to the DNS names. 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 5-1. 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><code>cloud.horizon.vmware.com</code></td>
</tr>
<tr>
<td>EU_CENTRAL_1 or Europe</td>
<td><code>cloud-eu-central-1.horizon.vmware.com</code></td>
</tr>
<tr>
<td>AP_SOUTHEAST_2 or Australia</td>
<td><code>cloud-ap-southeast-2.horizon.vmware.com</code></td>
</tr>
<tr>
<td>PROD1_NORTCENTRALUS2_CPI or USA-2</td>
<td><code>cloud-us-2.horizon.vmware.com</code></td>
</tr>
<tr>
<td>PROD1_NORTHEUROPE_CPI or Europe-2</td>
<td><code>cloud-eu-2.horizon.vmware.com</code></td>
</tr>
<tr>
<td>PROD1_AUSTRALIAEAST_CPI or Australia-2</td>
<td><code>cloud-ap-2.horizon.vmware.com</code></td>
</tr>
<tr>
<td>Japan</td>
<td><code>cloud-jp.horizon.vmware.com</code></td>
</tr>
<tr>
<td>UK</td>
<td><code>cloud-uk.horizon.vmware.com</code></td>
</tr>
<tr>
<td>Europe-3</td>
<td><code>cloud-de.horizon.vmware.com</code></td>
</tr>
<tr>
<td>Source</td>
<td>Destination (DNS name)</td>
</tr>
<tr>
<td>-----------------------</td>
<td>------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| Horizon Cloud Connector | cloud.horizon.vmware.com plus one of the following names, depending on which regional control plane instance is specified in your Horizon Cloud tenant account. The regional instance is set when the account is created, as described in Deployments and Onboarding to Horizon Cloud for Microsoft Azure and Horizon Pods.  
- 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  
- cloud-uk.horizon.vmware.com  
- cloud-de.horizon.vmware.com | 443  | TCP       | Regional control plane instance. Note: In addition to the regional instance as stated below, reachability to cloud.horizon.vmware.com is required by Horizon Cloud Connector for all tenant accounts.  
- 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  
- United Kingdom: cloud-uk.horizon.vmware.com  
- Germany: cloud-de.horizon.vmware.com |

**Note** (Horizon Cloud Connector 2.0 and later)  
This requirement applies to the primary node only.  
Depending on which regional control plane is specified in your Horizon Cloud account:  
- North America: kinesis.us-east-1.amazonaws.com

**Note**  
443  
TCP  
Cloud Monitoring Service (CMS)
<table>
<thead>
<tr>
<th>Source</th>
<th>Destination (DNS name)</th>
<th>Port</th>
<th>Protocol</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Horizon Cloud Connector</td>
<td>Europe, Germany: kinesis.eu-central-1.amazonaws.com</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Australia: kinesis.ap-southeast-2.amazonaws.com</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Japan: kinesis.ap-northeast-1.amazonaws.com</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>United Kingdom: kinesis.eu-west-2.amazonaws.com</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>*.digicert.com</td>
<td>80, 443</td>
<td>HTTP, HTTPS</td>
<td>CRL or OCSP queries used to obtain validation from the certificate authority, DigiCert</td>
</tr>
<tr>
<td></td>
<td>If your organization discourages the use of wildcards in allowable DNS names, you can allow specific names instead. For example, at the time of this writing, the specific DNS names required for certificate validation are:</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>ocsp.digicert.com</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>crl3.digicert.com</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>crl4.digicert.com</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td><a href="http://www.digicert.com/CPS">www.digicert.com/CPS</a></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>These DNS names are determined by DigiCert and subject to change. For instructions on how to obtain the specific names required by your certificates, refer to VMware Knowledge Base (KB) article 79859.</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>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>
| Horizon Cloud Connector | One of the following names, depending on which regional control plane instance is specified in your Horizon Cloud tenant account. The regional instance is set when the account is created, as described in Deployments and Onboarding to Horizon Cloud for Microsoft Azure and Horizon Pods.  
- connector-azure-us.vmwarehorizon.com  
- connector-azure-eu.vmwarehorizon.com  
- connector-azure-aus.vmwarehorizon.com  
- connector-azure-jp.vmwarehorizon.com  
- connector-azure-uk.vmwarehorizon.com  
- connector-azure-de.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  
- United Kingdom: connector-azure-uk.vmwarehorizon.com  
- Germany: connector-azure-de.vmwarehorizon.com |
<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>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>
Table 5-2. Horizon Cloud Connector Ports (continued)

<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>SDK endpoint of the vCenter Server,</td>
<td>443</td>
<td>TCP</td>
<td>This optional port configuration is required for use by the automated update feature. The automated update feature is deactivated by default and is only enabled on a per-pod basis by request. See Configure Automated Updates of the Horizon Cloud Connector Virtual Appliance.</td>
</tr>
<tr>
<td>Horizon Cloud Connector</td>
<td>SDK endpoint of the vCenter Server,</td>
<td>443</td>
<td>HTTPS</td>
<td>This optional port configuration is required for use by the Horizon Image Management Service. You only need to configure this port and protocol if the Horizon Image Management Service feature is enabled for your tenant account. See Managing Horizon Images from the Cloud.</td>
</tr>
</tbody>
</table>

Horizon Pod and Horizon Cloud Connector - Preparing to Onboard to Control Plane Services

When onboarding a Horizon pod to use control plane services, ensure you have the following items in place before you deploy the Horizon Cloud Connector appliance and run the onboarding wizard.

- Verify you have met the prerequisites described in Chapter 4 VMware Horizon Pods with Horizon Cloud Control Plane - Requirements Checklist - Appropriately Updated Starting from the v2204 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 and that account has a Horizon subscription license associated with it. This account is needed to run the Horizon Cloud Connector onboarding workflow to pair the pod with your specific service tenant, and for logging in to the cloud-based administrative console to perform administrative tasks, including adding additional administrators to your tenant.

- That My VMware account has been authorized to log in to your service tenant. You might be the initial subscriber to the service or the initial subscriber has added your My VMware email as an administrator to the tenant or an existing administrator in your organization added your My VMware email as an administrator to the tenant. All those cases would make for an authorized account. To see if your My VMware account is authorized to log in to your service tenant, go to https://cloud.horizon.vmware.com and enter your My VMware account credentials. If the system logs you in, then your account is authorized. If not, you will have to ask one of your organization’s tenant administrators to log in to the service tenant and add you or file a non-technical support request in Customer Connect using the steps in KB article 2006985 to request your addition to your organization’s existing tenant record.

- The Horizon Cloud Connector binary component must be downloaded from customerconnect.vmware.com, from the row labeled Horizon Cloud Connector located within the page URL https://customerconnect.vmware.com/downloads/info/slug/desktop_end_user_computing/vmware_horizon_service/1_x

- 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 appliance.

  **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 appliance in your pod’s 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.
- Decide on a strong root password for the virtual appliance. For all-in-SDDC deployments, the OVF deployment wizard will prompt for a password that contains a minimum of eight characters with one capital, one numeric, and one special character. For federated deployments, the password must adhere to the requirements of the native cloud platform.

**Important** All-in-SDDC deployments will involve using the OVF deployment wizard to deploy the appliance into the VMware SDDC. 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 7.10 or later. Version 7.10 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.

- When deploying the Horizon Cloud Connector appliance into a VMware SDDC environment, you must deploy it using vSphere Client or vSphere Web Client. Do not deploy the appliance directly into an 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.
- 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, 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 its deployed 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 configuration portal 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 configuration portal, 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:

```bash
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.

Old, Deprecated Versions

New deployments of Horizon Cloud Connector are supported using versions \( N, N-1, N-2 \) where \( N \) is the latest Horizon Cloud Connector version. Prior versions are deprecated for use. Existing deployments are expected to update to the same versions. For the numeric of the latest version, see the Release Notes.

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 Horizon Service Deployments and Onboarding Pods — Horizon Pods and Horizon Cloud Pods on Microsoft Azure.

You use the steps here to connect existing Horizon pods 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 4 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 Horizon Service Deployments and Onboarding Pods — Horizon Pods and Horizon Cloud Pods on Microsoft Azure, 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 VMware Customer Connect 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, your deployment should use the latest version of Horizon Cloud Connector that is available at VMware Customer Connect and which is compatible with the pod’s Horizon Connection Server software version. For the compatibility matrix between Horizon Cloud Connector and Horizon Connection Server, visit VMware Product Interoperability Matrix and check interoperability between the two solution names listed as VMware Horizon Cloud Connector and VMware Horizon.

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 image from the appropriate my.vmware.com Downloads area.
4. Download the Horizon Cloud Connector image using the link in the welcome email.
5. Deploy that appliance into the pod’s environment using a static IP address. When the virtual appliance deployment process is completed, power on the virtual appliance.
6. Obtain the 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 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 you obtained earlier 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 [Horizon Pod and Horizon Cloud Connector - Preparing to Onboard to Control Plane Services](#).

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 Horizon Pod and Horizon Cloud Connector - Preparing to Onboard to Control Plane Services, Horizon Cloud Connector Known Considerations, and Modifying Proxy Settings for Horizon Cloud Connector 1.6 or Later.

Procedure

1 Download and Deploy the Horizon Cloud Connector into Your Pod's Environment
   To download and deploy the Horizon Cloud Connector, follow the instructions in the subtopic that applies to your Horizon pod deployment architecture.

2 Verify the Horizon Pod and Virtual Appliance Are Ready to Be Paired with Horizon Control Plane
   In this step in the workflow of onboarding a Horizon pod to Horizon Control Plane services, 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 with the pairing process.

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.

4 Horizon Cloud Connector 2.0 and Later - Add a Worker Node to a Horizon Cloud Connector Cluster
   To support service-level fault tolerance for Horizon Cloud Connector, create a dual-node cluster by adding a worker node to the cluster containing the primary node. The worker node contains a replica of the Horizon Cloud Connector application services.

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, activate or deactivate 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 must stay open all the time to provide for the communication to the cloud control plane for various purposes, including syncing your up-to-date subscription licensing information to the pod. Use of the Horizon subscription licenses relies on an operational communication chain between the cloud control plane, the Horizon Cloud Connector instance, and the pod paired with that Horizon Cloud Connector instance. If a link in that communication chain is not operational, such as if the Horizon Cloud Connector is powered off or a network disruption occurs, the cloud plane continues its attempts to sync the subscription licensing information along the communication chain according to a service-defined timespan before marking the pairing with this pod and its Horizon Cloud Connector as expired. During this service-defined timespan, the pod’s licensing remains valid and end-user connections work even though the communication chain between the cloud plane and the Horizon Cloud Connector and pod is not operational. The service provides this service-defined sync validity period to allow the pod’s licensed features to continue functioning properly as you make the communication chain operational again. If the communication chain’s non-operational state persists throughout the system-defined timespan, such that there is no successful sync by the end of the defined timespan, the cloud plane marks the pairing with this Horizon Cloud Connector and pod as expired. At that time, you must contact VMware Support for assistance. For more details, see Monitoring the Horizon Universal License. Throughout the time period, alerts and notifications in the Horizon Universal Console inform you of how much time is left before the end of the system-defined timespan, and the audit logs in the console include indications of what the communication chain’s issue might be, to help you identify what to do to fix it.

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 Your Pod's Environment**

To download and deploy the Horizon Cloud Connector, follow the instructions in the subtopic that applies to your Horizon pod deployment architecture.

**Note** For background information on Horizon pod deployment architectures, see Horizon Pod Deployment Architectures.
For this Horizon pod deployment architecture: | Follow this Horizon Cloud Connector deployment procedure:
---|---
- On-premises pod
- All-in-SDDC pod | On-premises and All-in-SDDC Horizon Pods: Download and Deploy the Horizon Cloud Connector into the Pod's vSphere Environment
Federated pod - Microsoft Azure cloud and Azure VMware Solution (AVS) | Horizon Pods - Federated Architecture with Azure VMware Solution: Download and Deploy the Horizon Cloud Connector into the Pod’s Environment
Federated pod - Google Cloud Platform and Google Cloud VMware Engine (GCVE) | Horizon Pods - Federated Architecture with Google Cloud VMware Engine: Download and Deploy the Horizon Cloud Connector into Your Pod’s Environment
Federated pod - Amazon Web Services EC2 and VMware Cloud on AWS | Horizon Pods - Federated Architecture with VMware Cloud on AWS: Download and Deploy the Horizon Cloud Connector into Your Pod’s Environment

**On-premises and All-in-SDDC Horizon Pods: Download and Deploy the Horizon Cloud Connector into the Pod’s vSphere Environment**

Follow these steps to download and deploy Horizon Cloud Connector into the vSphere infrastructure of a Horizon pod that is deployed on premises or into a cloud environment with all-in-SDDC architecture. Performing these steps results in the Horizon Cloud Connector virtual appliance deployed and running in your vSphere environment.

This procedure covers what to do when you have deployed your Horizon pod on premises or into a cloud environment with all-in-SDDC architecture and, therefore, must deploy Horizon Cloud Connector into the pod’s vSphere environment.

**Note** For background information on Horizon pod deployment architectures, see Horizon Pod Deployment Architectures.

If you are downloading Horizon Cloud Connector 2.0 or later, the following procedure explains how to deploy the virtual appliance’s primary node into your pod’s vSphere environment.

If you are downloading Horizon Cloud Connector 1.10, the following procedure explains how to deploy the virtual appliance into your pod’s vSphere environment.

**Note** Unless otherwise specified, the steps in the procedure apply to all versions of Horizon Cloud Connector. These steps use the term "virtual appliance" to refer to either the appliance’s primary node (Horizon Cloud Connector 2.0 and later) or the virtual appliance (Horizon Cloud Connector 1.10).

**Important** You must deploy Horizon Cloud Connector into your pod’s vSphere environment using vSphere Client. Do not deploy Horizon Cloud Connector directly into the ESXi host.

**Prerequisites**
- Verify that you have met the connector-related prerequisites described in Horizon Pod and Horizon Cloud Connector - Preparing to Onboard to Control Plane Services.
- 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.

- To communicate with the Horizon Cloud control plane, the Horizon Cloud Connector virtual appliance must reach the Internet. 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 Horizon Pod and Horizon Cloud Connector - Preparing to Onboard to Control Plane Services, Horizon Cloud Connector Known Considerations, and Modifying Proxy Settings for Horizon Cloud Connector 1.6 or Later.

- If you want to enable Secure Shell (SSH) access to the appliance with the recommended level of security before pairing the appliance with the pod, generate an SSH public key. You must register the SSH public key during the deployment of the appliance.

**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 VMware Customer Connect after you log in to customerconnect.vmware.com using your VMware Customer Connect account credentials.

   **Important** To have the latest product fixes, security fixes, and the most up-to-date features, ensure that the downloaded version is the most recent generally available version or later. Currently version 2.2.x is the latest generally available version. If you previously downloaded a Horizon Cloud Connector OVA with a version prior to 2.2.x, log in to customerconnect.vmware.com and obtain the most recent version for pairing your pod.

2. Using vSphere Client, 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.
3 In the **Customize template** wizard step, complete the required items and specify 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 virtual appliance.

**Note** Ensure that the new password 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 you can access the virtual appliance after it is deployed, verify that the password contains at least one special character.

b (Horizon Cloud Connector 2.0 and later) To deploy the primary node of the appliance, verify that the **Worker Node** option is deactivated. By default, this option is deactivated.

c To use SSH public key authentication for the *ccadmin* account, enter the SSH public key that you generated earlier.

For more information, see [Enable SSH Access to the Horizon Cloud Connector Using a Command Line Interface](#).

If you do not need SSH access to the appliance before pairing the appliance with the pod, you can skip this step. You can defer registering a public key and enabling SSH access until after the appliance is paired with the pod.

d Specify a static IP address for the virtual appliance.

Do not use IPv6 with the Horizon Cloud Connector virtual appliance. IPv6 is not supported.
In the **Network** section, the **Pod Network** and the **Service Network** fields are optional. These networks are used by Kubernetes within the VM and are not accessible outside the VM. They should be at the default value unless they overlap with the customer internal networks. By default in the Kubernetes CNI, subnet 192.168.240.0/21 is used to configure **Pod Network** and 192.168.236.0/23 is used for **Service Network**. If you are configuring a **Worker Node** these settings do not apply.

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.

- 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.*`

  If you leave the **No Proxy For** setting blank, the virtual appliance fetches the Connection Server host names that are provided by the administrator or which are discovered. The appliance discovers Connection Server host names by querying the pod. It configures those discovered hosts as implicit no-proxy hosts.

4. Using vSphere Client, power on the Horizon Cloud Connector appliance.

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

   - (Horizon Cloud Connector 2.0 and later) Allow up to 10 minutes for Kubernetes to complete its initialization process. During this time, the system displays a blue startup screen with the message "Configuring Horizon Cloud Connector (Primary) ..." After the initialization is complete, the system displays the blue console screen with the URL address of the primary node. You will load this URL in your browser for the onboarding workflow.

   - (Horizon Cloud Connector 1.10) Wait for the appliance's blue console screen to appear with the URL address of the virtual appliance. You will load this URL in your browser for the onboarding workflow.

The following screenshot is an example for a deployed appliance that has an address `https://10.92.245.255/`. 
6 Complete the steps in *Enable SSH Access to the Horizon Cloud Connector Using a Command Line Interface.*

7 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 proceeding to *Verify the Horizon Pod and Virtual Appliance Are Ready to Be Paired with Horizon Control Plane.*

**Horizon Pods - Federated Architecture with VMware Cloud on AWS: Download and Deploy the Horizon Cloud Connector into Your Pod's Environment**

Follow these steps to download and deploy the Horizon Cloud Connector appliance for a pod deployment that uses the federated architecture with VMware Cloud on AWS. In the federated architecture, you must deploy Horizon Cloud Connector into the native Amazon Elastic Computer Cloud (EC2) infrastructure within your pod’s environment.

**Note** For background information on Horizon pod deployment architectures, see *Horizon Pod Deployment Architectures.*

The following is a high-level overview of the steps required to deploy Horizon Cloud Connector into the native Amazon EC2 infrastructure within your pod’s environment.

- Download the Horizon Cloud Connector file in VMDK format.
- Create an Amazon Simple Storage Service (S3) bucket on Amazon EC2 and upload the appliance VMDK file to that bucket.
- Create a custom image from the uploaded VMDK file.
- Create the Horizon Cloud Connector virtual machine (VM) instance from the custom image.

**Important** When deploying the Horizon Cloud Connector into the native Amazon EC2 infrastructure, the following Horizon Cloud services are deactivated by default. After deploying the appliance, you can optionally *Manually Activate Horizon Cloud Control Plane Services for Horizon Cloud Connector on Native Amazon EC2.*

- Cloud Monitoring Service
- Cloud Broker Client Service
- Image Locality Service
Prerequisites

- Verify that you have met the connector-related prerequisites described in Horizon Pod and Horizon Cloud Connector - Preparing to Onboard to Control Plane Services.

- 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 Horizon Pod and Horizon Cloud Connector - Preparing to Onboard to Control Plane Services, Horizon Cloud Connector Known Considerations, and Modifying Proxy Settings for Horizon Cloud Connector 1.6 or Later.

- You must use the AWS command-line for many of the steps. However, you can perform some of the deployment steps using either the AWS Management Console or the AWS Command Line Interface (CLI). For detailed information about working with an Amazon EC2 environment, see the Amazon Elastic Compute Cloud documentation at https://docs.aws.amazon.com/ec2/index.html. The steps that follow often suggest the specific type of Amazon Elastic Compute Cloud documentation to reference.

Procedure

1. Download the Horizon Cloud Connector disk image using the link provided in your subscription email.

   **Note** To support deployment of the appliance into an Amazon EC2 environment, you must download version 2.0 or later of the Horizon Cloud Connector disk image.

   The Horizon Cloud Connector disk image is available as a VMDK file after you log in to my.vmware.com using your My VMware account credentials. Download the VMDK file to your local system.

   Before uploading the disk image file to your Amazon EC2 environment, you must first create an Amazon S3 bucket.

2. Create an Amazon S3 bucket in your Amazon EC2 environment. For detailed instructions, refer to the Amazon Elastic Compute Cloud documentation.

3. Upload the downloaded VMDK file to your Amazon S3 bucket. You can perform this step using either the AWS Management Console or the AWS command-line interface (CLI).

   (AWS Management Console) Log in to the AWS Management Console for your Amazon EC2 environment. Go to the S3 service, select the bucket that you created earlier, and upload the VMDK file to that bucket.
- (AWS CLI) Access the AWS CLI and run the following command.

```
aws s3 cp <file-path-to-VMDK-file> <S3URI>
```

See the Amazon Elastic Compute Cloud documentation for details about running the `cp` command.

In the AWS Management Console, the VMDK file is listed in the **Objects** tab.
4 Create a service role and a policy and attach the policy to the role.
   
a Create the first of three new JSON files required for this procedure.  
   The purpose of this particular JSON file is to store the service-role information. Name the file as you wish. In this procedure, the example filename for this file is trust-policy.json.

b Create a service role with a name of your choice and store the role information in the new JSON file.
   For example, using the CLI, run a command such as the following.  
   The following command is a generic example.

   ```bash
   aws iam create-role --role-name <role-name> --assume-role-policy-document <file-path>
   ```
   The following example command replaces the placeholder `<role-name>` with the specific example `vmimport` and the placeholder `<file-path>` with the specific example `trust-policy.json`.

   ```bash
   aws iam create-role --role-name vmimport --assume-role-policy-document file://trust-policy.json
   ```
   See the Amazon Elastic Compute Cloud documentation for details about running the `create-role` command.
   The following text is as an example of the contents of the JSON file after you run the preceding command.

   ```json
   {
   "Version":"2012-10-17",
   "Statement":[
   {
   "Sid":"",
   "Effect":"Allow",
   "Principal":{
   "Service":"vmie.amazonaws.com"
   },
   "Action":"sts:AssumeRole",
   "Condition":{
   "StringEquals":{
   "sts:ExternalId":"vmimport"
   }"sts:ExternalId":"vmimport"
   }
   }
   ]
   }
   ```
c Create the second of three new JSON files required for this procedure.

Provide the name of bucket name in which you will upload the VMDK file, such as `<bucket-name>`, the name used in the example that follows.

The purpose of this particular JSON file is to attach a new policy with the new role. Name the file as you wish. In this procedure, the example filename for this file is `role-policy.json`.

The following text is as an example of the contents of the example `role-policy.json` file.

```json
{
   "Version":"2012-10-17",
   "Statement":[
   {
      "Effect":"Allow",
      "Action": [
          "s3:ListBucket",
          "s3:GetBucketLocation"
      ],
      "Resource": [
          "arn:aws:s3:::<bucket-name>"
      ]
   },
   {
      "Effect":"Allow",
      "Action": [
          "s3:GetObject"
      ],
      "Resource": [
          "arn:aws:s3:::<bucket-name>/*"
      ]
   },
   {
      "Effect":"Allow",
      "Action": [
          "ec2:ModifySnapshotAttribute",
          "ec2:CopySnapshot",
          "ec2:RegisterImage",
          "ec2:Describe*"
      ],
      "Resource": "*"
   }
   ]
}
```

d Create a policy, attach it to the new role, and store them in the newly created JSON file.

For example, using the CLI, run a command such as the following.

The following command is a generic example.

```
aws iam put-role-policy --role-name <role-name> --policy-name <policy-name> --policy-document <file-path>
```
The following specific example replaces the placeholder `<role-name>` with a specific example of a policy named `vmimport`, the placeholder `<policy-name>` with the specific example of the previously named role, also named `vmimport`, and the placeholder `<file-path>` with the specific example of the previously named JSON file, `role-policy.json`.

```
aws iam put-role-policy --role-name vmimport --policy-name vmimport --policy-document file://role-policy.json
```

See the Amazon Elastic Compute Cloud documentation for details about running the `put-role-policy` command.
5 Import a snapshot from the imported VMDK file.

a Create the third of three new JSON files required for this procedure.

Include the following information in the file.

- The bucket name, such as `<bucket-name>`, which is used in the example that follows.
- The filename of the VMDK file that you uploaded to your Amazon S3 bucket, such as `<vmdk-file-name-uploaded-to-S3>`, which is used in the example that follows.

The purpose of this particular JSON file is to store the snapshot of the imported VMDK file. Name the file as you wish. In this procedure, the example filename for this file is `container.json`.

The following text is an example of the contents of the `container.json` file.

```json
{
    "Description": "Adapter-VM",
    "Format": "vmdk",
    "UserBucket": {
        "S3Bucket": "<bucket-name>",
        "S3Key": "<vmdk-file-name-uploaded-to-S3>"
    }
}
```

b Run the command to import the snapshot from the imported VMDK file to the newly created JSON file.

Using the CLI, run the following type of command.

```
aws ec2 import-snapshot --role-name <role-name> --description <description> --disk-container <file-path>
```

See the Amazon Elastic Compute Cloud documentation for details about running the `import-snapshot` command.

The following command is a specific example of the `import-snapshot` command, where the `role-name` parameter is optional and not used, the description is "Adapter-VM", and the container filename is `container.json`.

```
aws ec2 import-snapshot --description "Adapter-VM" --disk-container file://container.json
```

The `import-snapshot` command can take several minutes to complete. However, after you run the command, the command creates output, which includes an `ImportTaskId` line that you can use to track the progress of the task. The following output is an example.

```json
{
    "ImportTaskId": "import-snap-05b4c84af4xxxxxxx",
    "Description": "Adapter-VM",
    "SnapshotTaskDetail": {
        "StatusMessage": "pending",
        "UserBucket": {
```
To track the progress of the `import-snapshot` task and obtain the snapshot ID, run the following command.

```bash
aws ec2 describe-import-snapshot-tasks --import-task-ids <import-task-id>
```

Replace the `<import-task-id>` placeholder with the value listed in the `import-snapshot` command output. The example value listed in the preceding example output is `import-snap-05b4c84af4xxxxxxx`. See the Amazon Elastic Compute Cloud documentation for details about running the `describe-import-snapshot-tasks` command.

The `describe-import-snapshot-tasks` command provides output that indicates the progress of the `import-snapshot` task and, when the task is complete, provides the snapshot ID, which is required to register the image. For example.

- "Progress": "43". A line in the output such as this line indicates the percentage of progress for the `import-snapshot` task. In this example, the task is 43% complete.
- "Status": "completed". A line in the output such as this line indicates that the `import-snapshot` task is complete.
- "SnapshotId": "snap-06d42e043bxxxxxxx". A line in the output such as this line is included when the task is complete. For this example, the snapshot ID is `snap-06d42e043bxxxxxxx`.

7 Take note of the snapshot ID from the `describe-import-snapshot-tasks` command output.

8 To register the snapshot image, run the `register-image` command.

```bash
aws ec2 register-image --region us-west-2 --name <image-name> --architecture x86_64 --root-device-name '/dev/sda1' --virtualization-type hvm --ena-support --block-device-mappings DeviceName=/dev/sda1,Ebs={SnapshotId=<SnapshotId>}
```

Where you must provide responses specific to your deployment for each option, such as the `--region`, `--architecture`, and others. See the Amazon Elastic Compute Cloud documentation for details about running the `register-image` command.

The following information is specific to the `--name` option and the `SnapshotId` parameter.

- `--name` - provide a name for the image according to the constraints for the string.
- **SnapshotId** - provide the snapshot ID from the `describe-import-snapshot-tasks` command output.

The `register-image` command provides output that includes the ID of the Amazon Machine Image (AMI). The following example is a typical `register-image` output.

```
{
  "ImageId": "ami-0721ee000321c4685"
}
```

The AMI indicated in the `register-image` command output also appears in the AWS Management Console among the list of AMIs.

9. To support the creation and configuration of the Horizon Cloud Connector AMI instance, prepare a startup script similar to the following example.

```
#!/bin/bash
/usr/bin/python3 /opt/vmware/bin/configure-adapter.py --sshEnable
sudo useradd ccadmin
echo -e 'password
password' | passwd ccadmin
echo 'cs_ip cs_fqdn' >> /etc/hosts
```

In the example, the script supports the following configurations:

- Enablement of SSH access to the Horizon Cloud Connector appliance.
- Creation of a `ccadmin` user account on the appliance with the defined password (`password`). Ensure that you define a strong password. Strong passwords are at least 8 characters and must include one or more numbers, upper and lower case letters, and special characters.
- Resolution of the Connection Server host name (`cs_fqdn`) to the Connection Server IP address (`cs_ip`).

You must add this script to the user-data in the next step where you launch the Horizon Cloud Connector AMI instance.

10. Launch the AMI instance for Horizon Cloud Connector.

**Important** To ensure that the instance provides sufficient capabilities, use model c5.2xlarge or greater.

You can launch an instance using the AWS Management Console or the CLI. In either case, use the ID of the Amazon Machine Image (AMI) provided in the `register-image` command output and add the startup script that you prepared in the previous step to the user-data.

**Important** You must add the startup script at this time because the user-data is only executed at the first boot sequence of the AMI instance.

To use the CLI, see the Amazon Elastic Compute Cloud documentation for details about running the `run-instances` command.
To use the AWS Management Console, see the Amazon Elastic Compute Cloud documentation for details, such as for launching an instance using the launch instance wizard.

If you choose to launch the instance using the AWS Management Console, locate the new AMI by the Image ID, select the AMI, and click **Launch**. You can then continue through the wizard providing the specifics for your deployment.

11 After the Horizon Cloud Connector AMI starts up, edit the configuration of the AMI instance and remove the startup script.

**What to do next**

Continue with the pod onboarding workflow by following the steps in [Verify the Horizon Pod and Virtual Appliance Are Ready to Be Paired with Horizon Control Plane](#). Then proceed to [Complete Pairing the Horizon Pod with Horizon Cloud Using the Horizon Cloud Connector Configuration Portal](#).

**Note** When you want to activate one or more of the cloud-plane services that are deactivated by default, you should activate them prior to completing the pairing of the pod with the cloud plane. See [Manually Activate Horizon Cloud Control Plane Services for Horizon Cloud Connector on Native Amazon EC2](#).

**Horizon Pods - Federated Architecture with Azure VMware Solution: Download and Deploy the Horizon Cloud Connector into the Pod's Environment**

Follow these steps to download and deploy the Horizon Cloud Connector appliance for a pod deployment that uses the federated architecture with Azure VMware Solution (AVS). In the federated architecture, you must deploy Horizon Cloud Connector into the native Microsoft Azure infrastructure of your pod's environment.

**Note** For background information on Horizon pod deployment architectures, see [Horizon Pod Deployment Architectures](#). For detailed information about working with an AVS environment, see the Microsoft Azure documentation at [https://docs.microsoft.com/en-us/azure/azure-vmware/](https://docs.microsoft.com/en-us/azure/azure-vmware/).

The following is a high-level overview of the steps required to deploy Horizon Cloud Connector into the native Azure infrastructure within your pod's environment:

- Download the Horizon Cloud Connector VHD file.
- Create an Azure storage container and upload the appliance VHD to that storage container.
- Create a virtual machine (VM) image from the uploaded VHD.
- Create the Horizon Cloud Connector VM from the VM image.

**Prerequisites**

- Verify that you have met the connector-related prerequisites described in [Horizon Pod and Horizon Cloud Connector - Preparing to Onboard to Control Plane Services](#).
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 Horizon Pod and Horizon Cloud Connector - Preparing to Onboard to Control Plane Services, Horizon Cloud Connector Known Considerations, and Modifying Proxy Settings for Horizon Cloud Connector 1.6 or Later.

To locate the required VM sizing, see Chapter 4 VMware Horizon Pods with Horizon Cloud Control Plane - Requirements Checklist - Appropriately Updated Starting from the v2204 Service Release

Procedure

1. Download the Horizon Cloud Connector disk image using the link provided in your subscription email.

The Horizon Cloud Connector disk image is available as a VHD file compressed into a ZIP package after you log in to my.vmware.com using your My VMware account credentials. Download and extract the VHD file to your local system.

**Important** To enable the most up-to-date features, download version 1.10 or later of the Horizon Cloud Connector disk image. Version 1.10 or later supports all Horizon Cloud features and services except for automated updates of Horizon Cloud Connector.

The following features and services are not supported for Horizon pods in AVS when paired with Horizon Cloud Connector 1.9 or earlier:

- Automated updates of Horizon Cloud Connector
- Universal Broker and multi-cloud assignments
- Cloud Monitoring Server (CMS)
- Horizon Image Management Service

Before uploading the disk image file to your AVS environment, you must first create an Azure storage container and share it using a shared access signature.
In the Azure portal, navigate to your storage account and create a storage container for the VHD file. For more information, see https://docs.microsoft.com/en-us/azure/storage/common/storage-account-overview.

During the creation of the shared access signature, a SAS token is generated. You must construct a storage account URL for the disk image file by appending the SAS token to the storage container URL.

a. Open the storage container by navigating to Storage Account > Properties > URL. Take note of the storage container URL for the next steps.

b. Create a shared access signature. Navigate to Storage Account > Shared access signature > Select Resource types and Generate SAS and Connection String. Take note of the generated SAS token for the next steps.

c. Construct the storage account URL using this format:

<StorageContainerPath>/HorizonCloudConnectorDiskImageName.vhd<SAS-Token>

The following is an example of a storage account URL:

https://azurestorage1.blob.core.windows.net/vmware/horizon-cloud-connector-1.8.0.0-16488286.vhd?sv=2020-01-01:ss-bfqt&st=2020-01-01T12:00:00Z&sp=rwdlapx&se=2020-01-01T06:00:00Z&spr=https&sig=dUPul74l4K0ah%2FdoCpaTTjY4t2Ja8kB%3D

3. Upload the disk image file to the storage account URL that you created.

a. Download and install the AzCopy utility on the local system where you extracted the VHD file containing the Horizon Cloud Connector disk image. For more information about the AzCopy utility, see https://docs.microsoft.com/en-us/azure/storage/common/storage-use-azcopy-v10.

b. To upload the VHD file, run the following command in the AzCopy utility:

azcopy cp <Path to extracted VHD file> "<StorageAccountURL>" --blob-type PageBlob

The following shows an example of an upload command issued from a local Windows computer:

azcopy cp c:\horizon-cloud-connector-1.9.0.0-16488286.vhd "https://azurestorage1.blob.core.windows.net/vmware/horizon-cloud-connector-1.8.0.0-16488286.vhd?sv=2020-01-01:ss-bfqt&st=2020-01-01T12:00:00Z&sp=rwdlapx&se=2020-01-01T06:00:00Z&spr=https&sig=dUPul74l4K0ah%2FdoCpaTTjY4t2Ja8kB%3D" --blob-type PageBlob
Create a virtual machine (VM) image from the uploaded VHD file.

a. In the Azure portal, navigate to **Images** and create a new VM image. Enter a name for the image, and specify the target location and resource group.

b. Specify the following options:
   - Set the **OS type** option to **Linux**.
   - Set the **VM generation** option to **Gen1**.

c. For the Storage blob, browse to the storage account and container that you created, and select the VHD file that you uploaded.

d. Click **Create** to create the VM image from the VHD file.

Deploy the Horizon Cloud Connector appliance by creating the appliance VM from the VM image.

a. In the Azure portal, open the VM image that you created in the previous step. Click **Create VM**.

b. Specify the following settings:
   - Enter a name for the new VM. This will be the hostname of the Horizon Cloud Connector appliance.
   - For **VM Sizing**, see Chapter 4 VMware Horizon Pods with Horizon Cloud Control Plane - Requirements Checklist - Appropriately Updated Starting from the v2204 Service Release

c. For the Administrator account, specify **ccadmin** as the user name. You must create this **ccadmin** user account in order to allow SSH access to the appliance.

d. For SSH Access, specify the **SSH Public Key** authentication method.

   **Note** Both SSH Public Key and Password authentication methods are supported. However, SSH Public Key provides stronger security and is the preferred method.

e. For Firewall settings, configure the following ports:
   - Port 443 for HTTPS
   - Port 22 for SSH

   If you intend to configure a firewall and proxy server for the appliance, you must also configure the appliance to allow certain public URLs. For more information, see **DNS, Ports, and Protocols Requirements When Using Horizon Cloud Connector and a Horizon Pod**.
For Network settings, specify a public IP Address allocation if you need to allow access to appliance over a public network. Also, specify the public inbound ports for HTTPS and SSH.

Navigate to VM Properties and take note of the IP address and FQDN of the appliance VM. You will need this information later to access the browser-based Horizon Cloud Connector configuration portal.

If your environment requires use of an HTTP proxy server for your virtual appliances to access the Internet, configure proxy-related settings for the appliance, as described in Modifying Proxy Settings for Horizon Cloud Connector 1.6 or Later.

Configure the required certificates, as described in Configure a CA-Signed Certificate for the Horizon Cloud Connector Virtual Appliance.

Complete the steps in Enable SSH Access to the Horizon Cloud Connector Using a Command Line Interface.

If you want to use a fully qualified domain name (FQDN) for the Horizon Cloud Connector 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.

What to do next

Continue with the pod onboarding workflow by following the steps in Verify the Horizon Pod and Virtual Appliance Are Ready to Be Paired with Horizon Control Plane. Then proceed to Complete Pairing the Horizon Pod with Horizon Cloud Using the Horizon Cloud Connector Configuration Portal.

Horizon Pods - Federated Architecture with Google Cloud VMware Engine: Download and Deploy the Horizon Cloud Connector into Your Pod's Environment

Follow these steps to download and deploy the Horizon Cloud Connector appliance for a pod deployment that uses the federated architecture with Google Cloud VMware Engine (GCVE). In the federated architecture, you must deploy Horizon Cloud Connector into the native Google Cloud Platform (GCP) infrastructure of your pod’s environment.

Note For background information on Horizon pod deployment architectures, see Horizon Pod Deployment Architectures. For detailed information about working with a GCVE environment, see the Google Cloud documentation at https://cloud.google.com/vmware-engine/docs.

The following is a high-level overview of the steps required to deploy Horizon Cloud Connector into the native GCP infrastructure within your pod’s environment:

- Download the Horizon Cloud Connector TAR file.
- Create a Google Cloud Storage bucket and upload the appliance TAR to that bucket.
- Create a custom image from the uploaded TAR file.
- Create the Horizon Cloud Connector virtual machine (VM) instance from the custom image.
Prerequisites

- Verify that you have met the connector-related prerequisites described in Horizon Pod and Horizon Cloud Connector - Preparing to Onboard to Control Plane Services.
- 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 Horizon Pod and Horizon Cloud Connector - Preparing to Onboard to Control Plane Services, Horizon Cloud Connector Known Considerations, and Modifying Proxy Settings for Horizon Cloud Connector 1.6 or Later.
- You can perform some of the deployment steps using either the Google Cloud graphical user interface (GUI) or the Google Cloud command-line interface (CLI). To use the CLI, you must first install the required components on your local system:
  - The gsutil tool. For instructions, see the Google Cloud Storage documentation.
  - The Google Cloud SDK. For instructions, see the Google Cloud SDK documentation.

Procedure

1. Download the Horizon Cloud Connector disk image using the link provided in your subscription email.

   The Horizon Cloud Connector disk image is available as a TAR file compressed into a .GZ package after you log in to my.vmware.com using your My VMware account credentials. Download the TAR file to your local system.

   **Note**  To deploy the appliance into a GCVE environment, download version 1.10 or later of the Horizon Cloud Connector disk image.

   Before uploading the disk image file to your GCVE environment, you must first create a Google Cloud Storage bucket.

2. Create a Google Cloud Storage bucket in your GCVE environment. For detailed instructions, refer to the Google Cloud documentation.

3. Upload the downloaded TAR file to your Google Cloud Storage bucket. You can perform this step using either the Google Cloud graphical user interface (GUI) or the Google Cloud command-line interface (CLI).

   - (GUI) Log in to the Google Cloud Platform for your GCVE environment. Go to the **Cloud Storage** page, select the bucket that you created earlier, and upload the TAR file to that bucket.
(CLI) Open the gsutil console and run the following command.

```bash
gsutil cp <file-path-to-TAR-file> gs://<bucket-name>
```

4. Create a custom image from the uploaded TAR file.

- (GUI) In the Google Cloud Platform, go to the **Compute Engine > Images** page. Select the option to create an image. In the image creation page, specify **Cloud Storage** as the source and browse to the uploaded TAR file in your bucket. Specify other image properties as appropriate and then proceed to create the image.

   Verify that the new image appears in the **Images** list.

- (CLI) In the gsutil console, run the image creation command, similar to the following example.

```bash
gcloud compute --project <project-name> images create <image-name> --description <image-description> --source-uri <TAR-file-uri>
```

**Note** You can customize the command with appropriate parameters as needed. For detailed information, see the Google Cloud SDK reference documentation.

5. To support the creation and configuration of the Horizon Cloud Connector VM instance, prepare a startup script similar to the following example.

```bash
#!/bin/bash
/usr/bin/python3 /opt/vmware/bin/configure-adapter.py --sshEnable
sudo useradd ccadmin
echo -e 'password
password

password
password'
| passwd ccadmin
echo 'cs_ip cs_fqdn
|' >> /etc/hosts
```

In the example, the script supports the following configurations:

- Enablement of SSH access to the Horizon Cloud Connector appliance.
- Creation of a **ccadmin** user account on the appliance with the defined password (**password**).
- Resolution of the Connection Server host name (**cs_fqdn**) to the Connection Server IP address (**cs_ip**).

6. Create the Horizon Cloud Connector VM instance from the custom image. Ensure that you configure a minimum of **n2-standard-8** for the VM sizing or machine type.

- (GUI) In the Google Cloud Platform, go to the **Images** page, select the custom image that you created earlier, and select the option to create a VM instance. Specify a minimum of **n2-standard-8** for the VM sizing or machine type, designate the custom image as the boot disk, and add the startup script that you prepared earlier. Specify other VM properties as appropriate and then proceed to create the VM instance.

   Verify that the Horizon Cloud Connector VM appears in the list of VM instances.
In the gsutil console, run the instance creation command, similar to the following example.

```
gcloud compute --project <project-name> instances create <instance-name>  
  --zone <zone>  --machine-type <n2-standard-8-minimum>  --network <network>  
  --subnet <subnet>  --maintenance-policy <maintenance-policy>  --scopes <scope>  
  --image <custom-TAR-image>  --metadata startup-script=<startup-script>
```

**Note** You can customize the command with appropriate parameters as needed. For detailed information, see the Google Cloud SDK reference documentation.

After the Horizon Cloud Connector VM starts up, edit the configuration of the VM instance and remove the startup script.

**Important** You must remove the startup script from the instance to prevent the script from running every time the Horizon Cloud Connector reboots.

**What to do next**

Continue with the pod onboarding workflow by following the steps in Verify the Horizon Pod and Virtual Appliance Are Ready to Be Paired with Horizon Control Plane. Then 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 Using a Command Line Interface**

For Horizon Cloud Connector 2.0 and later, use these steps if you want to use a Secure Shell (SSH) connection with the primary node before pairing the appliance with the pod or if you want to enable SSH access to the worker node. For Horizon Cloud Connector 1.10 and earlier, use these steps to enable SSH access to the deployed appliance before pairing it with the pod.

If you do not require SSH access to Horizon Cloud Connector before pairing the appliance with the pod, you can wait to enable it until after the appliance is paired with the pod. See Enable or Deactivate SSH on the Horizon Cloud Connector Appliance Using the Configuration Portal in the Administration Guide.

**Prerequisites**

For Horizon pods on premises or in VMware Cloud on AWS, perform the following tasks.

- Verify that the Horizon Cloud Connector appliance is successfully deployed into your pod’s environment as described in the subtopics under Download and Deploy the Horizon Cloud Connector into Your Pod’s Environment, but not yet paired with the Connection Server.

- (Horizon Cloud Connector 1.9 and later) To use the recommended public key authentication for SSH access, generate an SSH public key and register the key during the appliance deployment. See On-premises and All-in-SDDC Horizon Pods: Download and Deploy the Horizon Cloud Connector into the Pod’s vSphere Environment.
Alternatively, if you do not register the public key during deployment, you can copy the public key to the appropriate keys file after enabling SSH access, as described in the steps later in this topic.

Important  Beginning with Horizon Cloud Connector 1.9, SSH access is no longer supported for the root user account. For improved security, SSH access is only supported for the ccadmin user account through public-key (strongly recommended) or password authentication.

You can still use the root account to perform non-SSH administrative tasks on the appliance.

Use the following steps to enable SSH access to the Horizon Cloud Connector for the ccadmin user. As a security best practice, it is strongly recommended that you configure an SSH public key for authenticating the ccadmin user to the appliance.

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

2  Set the password for the ccadmin account.

   ```bash
   passwd ccadmin
   ```

   Note  Ensure that the new password 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.

3  Enable SSH access by running the following command.

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

4  Configure public-key authentication using one of the following methods.

   - If you registered the SSH public key in the Customize template step of the appliance deployment wizard, public-key authentication is already configured and no additional steps are required.

     For more information, see On-premises and All-in-SDDC Horizon Pods: Download and Deploy the Horizon Cloud Connector into the Pod's vSphere Environment.

   - If you did not register the SSH public key during the appliance deployment, run the following command from the client system, replacing `<IP_appliance>` with the IP address of the Horizon Cloud Connector appliance. When prompted, enter the ccadmin password.

     ```bash
     ssh-copy-id ccadmin@<IP_appliance>
     ```
The `ssh-copy-id` command copies the public key to the `ccadmin` user’s `~/.ssh/authorized_keys` file.

**Note** If you do not configure public-key authentication, password credentials are used to authenticate the `ccadmin` user for SSH access. For improved security, it is strongly recommended that you use public-key authentication instead of password authentication for SSH access.

SSH access to the appliance is now enabled.

**Note** To run commands as a `ccadmin` user with elevated permissions, append the `sudo` prefix to the commands in an SSH session.

**Enable SSH Access to Horizon Cloud Connector 1.8 and Earlier for Horizon Pods on Premises or in VMware Cloud on AWS**

To open an SSH connection to Horizon Cloud Connector 1.8 or earlier, you must enable SSH access and log in as the `root` user.

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

2. Enable SSH access by running the following command.

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

SSH access to the appliance is now enabled.

**Deactivate SSH Access to Horizon Cloud Connector for Horizon Pods on Premises or in VMware Cloud on AWS**

If you need to deactivate SSH access to the appliance, use the following command:

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

**Enable SSH Access to Horizon Cloud Connector for Horizon Pods in Azure VMware Solution (AVS)**

1. In the Azure portal, navigate to the Horizon Cloud Connector VM. Start the Run command action and choose `RunPowerShellScript`.

2. Enable SSH access by running the following command.

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

SSH access to the appliance is now enabled.

3. If you deployed Horizon Cloud Connector 1.7 and want to use SSH public key authentication, run the following additional command.

   ```bash
   chmod 744 /home/ccadmin
   ```
Enable SSH Access to Horizon Cloud Connector for Horizon Pods in Google Cloud VMware Engine (GCVE)

To enable SSH access to Horizon Cloud Connector prior to pairing the appliance with a pod in GCVE, include the appropriate line in the startup script when creating the Horizon Cloud Connector VM instance. See Horizon Pods - Federated Architecture with Google Cloud VMware Engine: Download and Deploy the Horizon Cloud Connector into Your Pod's Environment.

What to do next

Proceed to Verify the Horizon Pod and Virtual Appliance Are Ready to Be Paired with Horizon Control Plane. Then continue to Complete Pairing the Horizon Pod with Horizon Cloud Using the Horizon Cloud Connector Configuration Portal. When pairing is successfully completed, the Horizon Cloud Connector web-based configuration portal will provide a toggle that you can use to deactivate SSH access for the appliance, or re-enable SSH if it was previously deactivated.

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.

  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:

```
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>--proxyHost</td>
<td>Host name or IP address of the HTTP proxy server</td>
</tr>
<tr>
<td>--proxyPort</td>
<td>Port number for the proxy connection</td>
</tr>
<tr>
<td>--noProxyFor</td>
<td>Hosts or network range configured to bypass the HTTP proxy. Use spaces to separate multiple values.</td>
</tr>
<tr>
<td>--proxySsl</td>
<td>Specifies whether to use SSL for the proxy connection. Allowed values are true or false.</td>
</tr>
<tr>
<td>--proxyUsername</td>
<td>User name for the HTTP proxy</td>
</tr>
<tr>
<td>--proxyPassword</td>
<td>Password for the HTTP proxy</td>
</tr>
<tr>
<td>--implicitNonProxyHosts</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. 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 --noProxyFor bypass the proxy.</td>
</tr>
</tbody>
</table>

**Example Script**

```bash
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.

**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.
- Ensure that the PEM file is generated with the private key instead of the passphrase.
- 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:
     ```bash
     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:
     ```bash
     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:
     ```bash
     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:
     ```bash
     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:
     ```bash
     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:
     ```bash
     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:
     ```bash
     cp /root/server.crt /opt/container-data/certs/hze-nginx/server.crt
     ```
   - (Horizon Cloud Connector version 1.3 or earlier) Use the following command:
     ```bash
     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 2.0 or later) Use the following command:
     
     kubectl rollout restart daemonset hze-nginx -n hze-system
   
   - (Horizon Cloud Connector version 1.4 through 1.10) 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 Update the SSL thumbprints in the welcome screen.

- (Horizon Cloud Connector version 2.0 or later) Use the following commands:
  
  ```
  /opt/vmware/bin/configure-welcome-screen.py
  /usr/bin/killall --quiet vami_login
  ```

- (Horizon Cloud Connector version 1.4 through 1.10) 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 the cloud control plane 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.
Horizon Cloud Connector 1.8 or 1.9 with Basic Feature Profile: Manually Activate Horizon Cloud Services

If you deployed Horizon Cloud Connector 1.8 or 1.9 and selected the Basic Feature profile in the deployment wizard, only the Horizon subscription license service is activated. To activate additional cloud-based services that the Horizon Cloud Connector provides, follow the steps described in this article. You run the commands using a Secure Shell session on the Horizon Cloud Connector appliance.

When a Horizon Cloud Connector 1.8 or 1.9 appliance is deployed with the Basic Feature profile selected, the following components are deactivated by default in the deployed appliance. You can use this procedure’s steps to manually activate one or more of these components. Each component supports the use of a specific cloud-plane service.

Connection Server Monitoring Service (CSMS)

This component supports the use of the Cloud Monitoring Service (CMS) with a Horizon pod. When you want the ability to use CMS when the Horizon Cloud Connector was deployed with the Basic Feature profile, you must use the steps below to activate this component. The CMS is described in the pages at Introducing the Cloud Monitoring Service's Unified Visibility and Insights.

Cloud Broker Client Service (CBCS)

This component supports the use of Universal Broker with a Horizon pod. You must activate this component if you want to use Universal Broker and configure multi-cloud assignments based on resources in your Horizon pod. The Universal Broker is described in the pages at System Architecture and Components of Universal Broker.

Image Locality Service (ILS)
This component supports the use of the Horizon Image Management Service with a Horizon pod. If you want to use the Horizon Image Management Service to track and manage system images from your Horizon pod, you must activate this Image Locality Service. The Horizon Image Management Service is described in the pages at Managing Horizon Images from the Cloud.

**Important** Observe the following guidelines:

- Use these steps only to activate services if you deployed Horizon Cloud Connector 1.8 or 1.9 and selected the **Basic Feature** profile during the appliance deployment.
- Do not perform these manual activation steps for any other versions of Horizon Cloud Connector. These steps do not apply to any other version.
- Do not perform these manual activation steps if you deployed version 1.8 or 1.9 with the **Full Feature** profile. In that scenario, the cloud-management services supported for use with Horizon pods are already activated and running by default.
- Once you have activated a service, do not attempt to deactivate it manually. Deactivating a service can produce unexpected results.
- If you decide to keep the CSMS service inactive in the Horizon Cloud Connector, to prevent **sync failed** messages from occurring, you should toggle off the CMS in your tenant account before you complete the pairing of the pod. See When Keeping the CSMS Service is Deactivated in the Horizon Cloud Connector, How to Prevent the Sync Failed Message.

**Procedure**

1. Navigate to the deployed Horizon Cloud Connector 1.8 or 1.9 appliance and configure the resource capacity required by the additional services that you want to activate.

<table>
<thead>
<tr>
<th>Additional Service to Activate</th>
<th>Minimum Resources</th>
</tr>
</thead>
<tbody>
<tr>
<td>Connection Server Monitoring Service (CSMS)</td>
<td>Total of 7 vCPUs, 8 GB memory (RAM), 40 GB datastore</td>
</tr>
<tr>
<td>Cloud Broker Client Service (CBCS)</td>
<td>Total of 6 vCPUs, 8 GB memory (RAM), 40 GB datastore</td>
</tr>
<tr>
<td>Image Locality Service (ILS)</td>
<td>Total of 6 vCPUs, 7.5 GB memory (RAM), 40 GB datastore</td>
</tr>
<tr>
<td>Two or more</td>
<td>Total of 8 vCPUs, 8 GB memory (RAM), 40 GB datastore</td>
</tr>
</tbody>
</table>
2 Activate and start each service that you want to use.
   
   a Open a Secure Shell (SSH) session on the Horizon Cloud Connector appliance, and log in as the root user.
   
   b Run the commands corresponding to each service that you want to activate.

<table>
<thead>
<tr>
<th>Service</th>
<th>Commands</th>
</tr>
</thead>
</table>
| Connection Server Monitoring Service (required for the Cloud Monitoring Service) | systemctl enable csms  
systemctl restart csms |
| Cloud Broker Client Service                                  | systemctl enable cbcs  
systemctl restart cbcs |
| Image Locality Service                                       | systemctl enable ils  
systemctl restart ils |

What to do next

If you activated the Connection Server Monitoring Service (CSMS), you must also prompt your tenant's Cloud Monitoring Service setting to synchronize the service to receive the monitoring data that the CSMS sends to it:

1 In your browser, log in to the Horizon Universal Console at cloud.horizon.vmware.com using your tenant login credentials, navigate to the Getting Started page, expand the page's General Setup section, and in the Cloud Monitoring Service row, click Edit.

2 Depending on the current state of the Cloud Monitoring Service toggle, perform one of the following steps:
   
   If you see that the toggle is switched off, switch it on.
   
   If you see that toggle is already switched on, first switch off the toggle and wait a few minutes. Then switch the toggle back on. This combination prompts the service to begin receiving the data from the newly activated CSMS.

Manually Activate Horizon Cloud Control Plane Services for Horizon Cloud Connector on Native Amazon EC2

When you deploy Horizon Cloud Connector into the native Amazon EC2 infrastructure, only the subscription license service is in an activated state. To activate additional cloud-based services
that the Horizon Cloud Connector provides, use a Secure Shell session on the Horizon Cloud Connector appliance as illustrated in the steps that follow.

**Important**

- If you decide to activate one or more of these services, perform the activation before you complete the pairing of the Horizon pod with Horizon Cloud. See Verify the Horizon Pod and Virtual Appliance Are Ready to Be Paired.
- If you decide to keep the CSMS service inactive in the Horizon Cloud Connector, to prevent sync failed messages from occurring, you should toggle off the CMS in your tenant account before you complete the pairing of the pod. See When Keeping the CSMS Service is Deactivated in the Horizon Cloud Connector, How to Prevent the Sync Failed Message.

You can optionally activate one or more of these services in the Horizon Cloud Connector.

**Connection Server Monitoring Service (CSMS)**

This component supports the use of the Cloud Monitoring Service (CMS) with a Horizon pod. When you want the ability to use CMS with this Horizon Cloud Connector deployment and its paired Horizon pod, you must use the steps below to activate this component. The CMS is described in the pages at Introducing the Cloud Monitoring Service's Unified Visibility and Insights.

**Cloud Broker Client Service (CBCS)**

This component supports the use of Universal Broker with a Horizon pod. You must activate this component if you want to use Universal Broker and configure multi-cloud assignments based on resources in the Horizon pod. The Universal Broker is described in the pages at System Architecture and Components of Universal Broker.

**Image Locality Service (ILS)**

This component supports the use of the Horizon Image Management Service with a Horizon pod. If you want to use the Horizon Image Management Service to track and manage system images from your Horizon pod, you must activate this component. The Horizon Image Management Service is described in the pages at Managing Horizon Images from the Cloud.

**Note** In the current release, use of Horizon Image Management Service (IMS) with Horizon deployments is supported only with these two deployment models: on-premises deployments and deployments with VMware Cloud on AWS using all-in-SDDC architecture. Use of IMS with Horizon deployments other than those two is currently unsupported. Therefore, although you can use the following steps to activate the Image Locality Service component in the Horizon Cloud Connector deployed into native Amazon EC2 for a federated deployment, actual use of the corresponding features is not currently supported in that federated deployment type.
Prerequisites

Confirm that the version of your deployed Horizon Cloud Connector VM meets the resource requirements as stated in Chapter 4 VMware Horizon Pods with Horizon Cloud Control Plane - Requirements Checklist - Appropriately Updated Starting from the v2204 Service Release.

Procedure

1. Navigate to the deployed Horizon Cloud Connector appliance in your Amazon EC2 infrastructure within your pod's environment.

2. Perform the activation steps for each service that you want to use.
   a. Open a Secure Shell (SSH) session on the Horizon Cloud Connector appliance, and log in as the root user.
   b. Create a backup of the /opt/container-data/conf/container-info file.

        ```
cp /opt/container-data/conf/container-info  /opt/container-data/conf/container-info-backup
        ```
   c. Open the /opt/container-data/conf/container-info file for editing.

        ```
vi /opt/container-data/conf/container-info
        ```
d Update the "enabled" flag from "n" to "y" for each of the services you want to activate:

Cloud Broker Client Service (CBCS), Connection Server Monitoring Service (CSMS), and
Image Locality Service (ILS).

```yaml
{
  "name": "cbcs",
  "namespace": "cbcs-system",
  "buildNumber": "1593-b1b5139",
  "enabled": "y",
  "firstboot": "y",
  "profile": "all",
  "imgUrl": "hcs-docker-local.artifactory.eng.vmware.com/hcs-broker/dev/websocket-client:1593-b1b5139"
},
{
  "name": "csms",
  "namespace": "csms-system",
  "buildNumber": "ultron-10",
  "enabled": "y",
  "firstboot": "y",
  "profile": "all",
  "imgUrl": "eucsupp-docker-local.artifactory.eng.vmware.com/cloudmonitoring/csms:ultron-10"
},
{
  "name": "ils",
  "namespace": "ils-system",
  "buildNumber": "9",
  "enabled": "y",
  "firstboot": "y",
  "profile": "all",
  "imgUrl": "hcs-docker-local.artifactory.eng.vmware.com/image-locality-service/ils-k8s-1/image-locality-service:9"
}
```

e Save the file.

f Run the appropriate command for each service you want to activate on the Horizon Cloud Connector.

Run the following command to activate the Cloud Broker Client Service.

```
kubectl apply -f /opt/vmware/docker-container/cbcs/charts/cbcs-component.yaml
```

Run the following command to activate the Cloud Monitoring Service.

```
kubectl apply -f /opt/vmware/docker-container/csms/charts/csms-component.yaml
```

Run the following command to activate the Image Locality Service.

```
kubectl apply -f /opt/vmware/docker-container/ils/charts/ils-component.yaml
```
What to do next

If you activated the Connection Server Monitoring Service (CSMS), you must also prompt your tenant's Cloud Monitoring Service setting to synchronize the service to receive the monitoring data that the CSMS will send to it after the pod is paired with the Horizon Cloud Connector.

1. In your browser, log in to the Horizon Universal Console at cloud.horizon.vmware.com using your tenant login credentials, navigate to the Getting Started page, expand the page's General Setup section, and in the Cloud Monitoring Service row, click Edit.

2. Depending on the current state of the Cloud Monitoring Service toggle, perform one of the following steps:
   - If you see that the toggle is switched off, switch it on.
   - If you see that toggle is already switched on, first switch off the toggle and wait a few minutes. Then switch the toggle back on. This combination prompts the service to begin receiving the data from the CSMS.

When Keeping the CSMS Service is Deactivated in the Horizon Cloud Connector, How to Prevent the Sync Failed Message

This page describes the best practice for when you want to keep the CSMS service deactivated in the Horizon Cloud Connector. All tenants in the cloud plane are created to have the CMS initially enabled by default and remain that way unless you explicitly toggle it off. If the CSMS service is in a deactivated state within a pod's Horizon Cloud Connector and the tenant account has the Cloud Monitoring Service (CMS) setting toggled on, a sync failed error is reported for that pod.

If you want to keep CSMS deactivated in the Horizon Cloud Connector, a best practice is to log in to the Horizon Universal Console and toggle off the CMS setting in your tenant account before you pair the pod with the Horizon Cloud Connector. The Cloud Monitoring Service (CMS) is one of the cloud-plane services provided to each tenant, while the CSMS service is a microservice that runs within the Horizon Cloud Connector to communicate with that CMS in the cloud plane. Some deployments of the Horizon Cloud Connector appliance will have the CSMS service deactivated by default, such as when you select use of the Basic Profile for version 1.9. In such deployments, you can optionally activate the CSMS service when you are ready to take advantage of the cloud-plane's Cloud Monitoring Service's features.

However, when the CSMS service is not running in the Horizon Cloud Connector at the same time when the CMS toggle is enabled for your tenant, a sync failed error is reported for that pod. In this use case where you want to leave the CSMS microservice inactive, to prevent having this sync failed message, you should toggle off the Cloud Monitoring Service in your tenant account before pairing the pod with the Horizon Cloud Connector.
To toggle off the Cloud Monitoring Service in your tenant account, log in to the Horizon Universal Console at cloud.horizon.vmware.com. If your tenant has no pods yet, the Getting Started page of the console should be visible to you right after you log in. The CMS toggle is located within the Cloud Monitoring Service row under the General Setup menu. Expand General Setup, click Edit within the Cloud Monitoring Service row, and toggle off the Cloud Monitoring Service. After it is toggled off, then you can proceed with pairing the pod with the CSMS microservice not running in the Horizon Cloud Connector and avoid the sync failed message at the same time.

Verify the Horizon Pod and Virtual Appliance Are Ready to Be Paired with Horizon Control Plane

In this step in the workflow of onboarding a Horizon pod to Horizon Control Plane services, 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 with the pairing process.

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 the control plane with your Horizon pod successfully. In addition, the tool checks whether:

- The configurations related to certificates and proxy settings are correct.
- Connectivity of Horizon Cloud Connector with the control plane and with Horizon 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 Your Pod's Environment, including the steps to Enable SSH Access to the Horizon Cloud Connector Using a Command Line Interface and if you want to activate any of the cloud-hosted services, the steps to manually activate those services.
- 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.

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

If the diagnostic tool discovers an issue that will prevent a successful pairing process, 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 process.

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

  This condition pertains to the control plane’s 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 the 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 process, because the initialization process requires connectivity to the control plane. Therefore, the FAIL condition is expected at this stage of the onboarding workflow. In a successful pairing process, 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.

**Important** To ensure the proper functioning of the Horizon subscription license and other cloud-hosted services for a pod, you must pair only one Horizon Cloud Connector appliance with that pod. The pairing of more than one appliance with a pod is allowed only under special circumstances such as appliance updates and troubleshooting procedures.

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 79859.

### Prerequisites

Verify that you have completed the required preparation steps:

- Verify that you have completed the steps in the applicable subtopics under Download and Deploy the Horizon Cloud Connector into Your Pod's Environment.

- Verify that you have completed the steps in Verify the Horizon Pod and Virtual Appliance Are Ready to Be Paired with Horizon Control Plane.

Also, verify that:

- The primary node (Horizon Cloud Connector 2.0 and later) or virtual appliance (Horizon Cloud Connector 1.10 and earlier) is powered on.
You have the URL for displaying the browser-based Horizon Cloud Connector configuration portal. This URL 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 Horizon Pod and Horizon Cloud Connector - Preparing to Onboard to Control Plane Services, 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 Horizon Pod and Horizon Cloud Connector - Preparing to Onboard to Control Plane Services, Horizon Cloud Connector Known Considerations, and Modifying Proxy Settings for Horizon Cloud Connector 1.6 or Later.

- 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.

- You have the following information to provide in the first setup step of the wizard:
  - FQDN of the Horizon Connection Server
  - Active Directory domain's NETBIOS name
  - Account name of an Active Directory user in that domain. You will type in the SAM type of user name. Do not type in a UPN (User Principal Name). This user account must meet the requirements as stated in the Horizon Pod and Horizon Cloud Connector Requirements section of the Horizon Pods with Horizon Cloud Control Plane Requirements Checklist.

Procedure

1. Get the URL for launching the web-based configuration portal.
   - (Horizon pods on premises or in VMware Cloud on AWS) Get the URL from the appliance’s blue console screen.
(Horizon pods in Azure VMware Solution) Get the URL by navigating to the appliance VM in the Azure portal VM Properties and noting the IP address or FQDN of the appliance VM. Construct the URL as follows: https://IP-address or https://FQDN/

2 Using a browser, navigate to the URL that you obtained in the previous step.

   **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**.

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.
Note At this point, the system detects whether the Horizon Cloud Connector environment is configured improperly. If this is the case, a message appears prompting you to proceed with the cleanup tasks required to correct the configuration.

4 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.

5 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 Active Directory domain name associated with this Connection Server.

7 After the domain name, type the SAM type of account name for an Active Directory user in that domain and its associated password and click **Connect**.
   - Type in the SAM type of user name, without the domain name portion. Do not type in a UPN (User Principal Name). Even though the UI field accepts UPN format and fails to reject it, subsequent onboarding tasks will fail if a UPN form is entered here.
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 Horizon Pod and Horizon Cloud Connector - Preparing to Onboard to Control Plane Services.

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. If this is the case, the page displays a message asking whether you want to perform a New Install action that deletes the existing pairing and pairs 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 Upgrade action that copies the existing appliance 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 next steps of this procedure.

Step 2 of the wizard appears.

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>Enter 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.</td>
</tr>
<tr>
<td>Description</td>
<td>Optional: Enter a description for this pod.</td>
</tr>
</tbody>
</table>

9  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.

When the system determines that 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 screen also shows the health status of activated cloud-hosted services. A gray dash icon indicates that the service is deactivated and therefore has no health status to display.

The following screenshot is an example of the congratulations screen.
You might see components that always appear in a deactivated state. Such components are planned for use in a future service release.

When the Connection Server is Located in a vSphere Infrastructure, the Horizon Cloud Connector vCenter Server Details Window Appears Automatically

When the Horizon Cloud Connector and its paired Connection Server are installed into a vSphere infrastructure of a VMware SDDC or on-premises, the Horizon Cloud Connector vCenter Server Details window automatically opens in front of the congratulations screen.

When the Horizon Cloud Connector and Connection Server are installed natively into a public cloud infrastructure — as in the federated deployment type — this feature does not apply.
The entered values are intended to support the automated updates of the Horizon Cloud Connector virtual appliance.

To close this window, you must enter vCenter Server and network details as described in the article Automated Updates of the Horizon Cloud Connector Virtual Appliance in the Administration Guide. The configuration of these details is mandatory, even though the automated update feature is an optional capability that is enabled on a per-pod basis.

If you close the Horizon Cloud Connector vCenter Server Details window without entering the mandatory details, you will encounter persistent warning messages until you click the Configure vCenter Server and Network Details button and enter the details.

The following screenshot shows an example of the window as it initially appears.
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 four 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.

(Horizon Cloud Connector 2.0 and later) To support service-level fault tolerance, deploy a worker node. See Horizon Cloud Connector 2.0 and Later - Add a Worker Node to a Horizon Cloud Connector Cluster.

Horizon Cloud Connector 2.0 and Later - Add a Worker Node to a Horizon Cloud Connector Cluster

To support service-level fault tolerance for Horizon Cloud Connector, create a dual-node cluster by adding a worker node to the cluster containing the primary node. The worker node contains a replica of the Horizon Cloud Connector application services.
To add a worker node to a cluster, you first use vSphere Client to deploy the worker node into your pod’s vSphere environment. Then you run commands to join the worker node to the cluster containing the primary node.

For an overview of nodes, clusters, and the fault tolerance and high availability features of Horizon Cloud Connector, see Horizon Cloud Connector Clusters, Node-Level High Availability, and Service-Level Fault Tolerance.

**Note** This release supports dual-node clusters, node-level high availability, and service-level fault tolerance only for appliances paired with the following types of pods:

- Horizon pods deployed on premises
- Horizon pods deployed in VMware Cloud on AWS with all-in-SDDC architecture

Horizon pods deployed in all other environments support single-node clusters consisting of a primary node only and do not support node-level high availability and service-level fault tolerance.

**Prerequisites**

Ensure that you have completed the following preliminary tasks:

- Deploy the primary node of the cluster and pair Horizon Cloud Connector with your Horizon pod, as described in Download and Deploy the Horizon Cloud Connector into the Pod's vSphere Environment.
- Verify that you have met the DNS, ports, and protocols requirements for ongoing operations between Horizon Cloud and the Horizon Cloud Connector virtual appliance.

**Procedure**

1. To deploy the worker node, follow the general procedure for deploying the Horizon Cloud Connector appliance as an OVF template, as described in Download and Deploy the Horizon Cloud Connector into the Pod's vSphere Environment. Ensure that you configure the following options.
   - In the Customize template wizard page, enable the Worker Node option. By default, the Worker Node option is deactivated, resulting in a primary node deployment.
   - Configure an SSH public key for the ccadmin user of the worker node. You must enable SSH access to the worker node in order to run the required commands later in this procedure.

2. After the deployment is complete, use vSphere Client to power on the worker node VM. On the blue console screen, make a note of the node's URL for launching the Horizon Cloud Connector configuration portal.

3. Enable SSH access to the worker node for the ccadmin account. See Enable SSH Access to the Horizon Cloud Connector Using a Command Line Interface.
4 Open an SSH session to the primary node in the cluster and run the following command, where `<WORKER_IP>` is the IP address of the worker node that you got earlier.

```
/opt/vmware/sbin/primary-cluster-config.sh -as <WORKER_IP>
```

5 At the prompt asking if you want to continue connecting to the worker node, type `yes`.

6 Allow the command to execute without interruption. At the end of the command output, look for lines similar to the following example, where `<PRIMARY_IP>` is the IP address of the primary node VM.

```
Please run the following command on worker node to join the cluster:
/opt/vmware/sbin/worker-cluster-config.sh -a <PRIMARY_IP> <TOKEN 1> <TOKEN 2>
```

Make a note of this cluster-join command.

7 Open an SSH session to the worker node and run the cluster-join command that you got in the previous step.

```
/opt/vmware/sbin/worker-cluster-config.sh -a <PRIMARY_IP> <TOKEN 1> <TOKEN 2>
```

8 After joining the worker node to the cluster, run the following command on the primary node VM to verify the new cluster membership.

```
kubectl get nodes -o wide
```

In the output returned by the command, verify that both the primary node and worker node appear as registered members of the cluster, identified by their IP addresses.
Horizon Cloud on Microsoft Azure Deployments - Key Characteristics

This documentation page introduces the key characteristics of Horizon Cloud on Microsoft Azure deployments. These deployments are also referred to as Horizon Cloud pods.

See also:

- Horizon Service Deployments and Onboarding Pods — Horizon Pods and Horizon Cloud Pods on Microsoft Azure for the overall introduction to what to do when your tenant account is newly created, the day-0 tasks prior to deploying your first pod, and the day-1 tasks of onboarding the pod.

Brief Introduction

Horizon Cloud pods are based on the Horizon Cloud pod-manager technology. This pod-manager technology runs solely in a Microsoft Azure subscription and does not require a VMware SDDC.

A Horizon Cloud on Microsoft Azure deployment is distinct from other Horizon deployment types, which involve Horizon Connection Server software components.

Deployments

You deploy Horizon Cloud on Microsoft Azure by deploying a pod-manager-based pod using the automated pod deployment wizard. You must have a subscription for cloud capacity in Microsoft Azure and then bring that subscription information to pair that cloud capacity with your Horizon Cloud tenant.

The wizard deploys the required VMware software components into your Microsoft Azure cloud subscription, creating the Horizon Cloud on Microsoft Azure deployment.

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.
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 VMware Customer Connect 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 manager instances.
- 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 which 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 deployer 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 deploy the pod initially without either type and add on later.

The system deploys the pod with high availability, having two pod manager VMs by default.
Figure 6-1. Illustration of the Horizon Cloud Pod Architecture where the Pod has 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.
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**.

**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.
## Useful Microsoft Azure References

<table>
<thead>
<tr>
<th>Reference</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</td>
<td>Use these articles to learn about how to configure VPNs in Microsoft Azure.</td>
</tr>
<tr>
<td>Planning and design for VPN Gateway</td>
<td></td>
</tr>
<tr>
<td>Create a Site-To-Site connection in the Azure portal</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 Azure Virtual Desktop?</td>
<td>Use this article to learn about Microsoft Azure 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 Azure 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</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</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>

This chapter includes the following topics:

- Get Started with a Simplified, Proof-of-Concept Horizon Cloud Service on Microsoft Azure Pod Deployment
- Horizon Cloud on Microsoft Azure - First Pod Deployment - High-Level Workflow
- Preparing to Deploy a Horizon Cloud Pod on Microsoft Azure
- Using the Horizon Universal Console to Perform an Automated Deployment of a Pod into Microsoft Azure
- Troubleshooting If You Encounter Pod Deployment or First-Time Domain Bind Issues

Get Started with a Simplified, Proof-of-Concept Horizon Cloud Service on Microsoft Azure Pod Deployment

This page describes the ingredients and step-by-step recipe for a pod to use in a simplified, kick-the-tires exploration of Horizon Cloud Service on Microsoft Azure deployment.

Brief Introduction

The intent of this page is to provide a step-by-step recipe that creates a smooth, simplified pod deployment suitable for kick-the-tires environments such as proof of concepts, home labs, pilots, trial environments, and the like.
Such environments are expected to be torn down after the kick-the-tires exploration is over.

This recipe here is solely for use with a single subscription, one basic VNet, and a local PoC Active Directory VM — anything else is out of scope.

For anything beyond this, please engage VMware Professional Services. Through their Delivery Specialist program, VMware Professional Services provides a design-build approach to implementation and onboarding tailored to your specific Horizon Cloud Service on Microsoft Azure deployment needs.

**Ingredients for a Simplified Initial Deployment**

This step-by-step recipe as written in this page was proved out by a college graduate using these minimal essential ingredients.

**Ingredients**

- One pay-as-you-go Azure subscription backed by an industry-standard credit card.
- West US 3 Azure regional location in that subscription.
- One basic, single VNet in that subscription, configured for 512 addresses (10.0.0.0/23).
- A local PoC Active Directory VM on that VNet, to satisfy the register Active Directory flow.
- Use the Azure built-in Contributor role for the app registration that we set up for Horizon Cloud Service for making its Azure API calls.

The West US 3 region was used in this recipe because, at the time of this writing, the West US 3 region satisfied these two goals of the PoC deployment: that region is geographically closest to the intended VDI end users and that region satisfied the deployment's Azure Managed PostgreSQL service and VM Family vCPUs requirements using the pay-as-you-go subscription.

**Simplifications**

- Because the gateway feature can be added onto the deployed pod later using Edit Pod, we simplified our recipe here by initially deploying with the gateway configuration toggles switched off.

  Doing this allows for successfully completing the initial deployment in parallel with obtaining the SSL certificate needed for the gateway configuration.

- To simplify the quota checking step, we omit checking for quota for the VM families that the system requires for golden images that use Windows 11 operating systems (OSes). The system requires a different VM model for Windows 11 golden images than for Windows 10 ones. To simplify, we omit the Windows 11 use case from our quota check.

**When You Use the Azure Portal**

The activities involved in preparing Microsoft Azure rely on the Azure Portal.
Wherever our recipe here refers to the Azure Portal, please understand that:

- Microsoft updates their interfaces occasionally over time.
- Microsoft also personalizes everyone's portal experience according to their account access and their portal settings.
- We do our best to keep both the screenshots here and the labels and names displayed in the Azure Portal up-to-date with Microsoft's changes.
- The screenshots in this page and the labels and names might not perfectly match what you see in the Azure Portal at a given point in time, due to how Microsoft rolls out updates and personalizes your portal experience.

This PoC page uses the term *pane* to refer to an area of the Azure Portal.

**Prepare Microsoft Azure Cloud**

For this PoC recipe, we must prepare our Microsoft Azure subscription with elements that the PoC Horizon Cloud on Microsoft Azure deployment needs before running the Add Pod wizard.

The screenshots contained in these sections illustrate what we saw for our pay-as-you-go subscription used to prove out this page's steps.

Details that you see in your specific Azure environment will look different, because Microsoft personalizes what you see and have access to.
Obtain Azure Subscription

The first activity for the PoC is obtaining an Azure subscription for the PoC deployment.

By its definition, a Horizon Cloud Service on Microsoft Azure deployment resides in a Microsoft Azure subscription that you provide.

As of this writing, Microsoft makes available these main types of Azure subscriptions: free types, pay-as-you-go, and enterprise-type subscriptions.

Currently, the pay-as-you-go and enterprise-type subscriptions are the ones that support having the quota levels that a Horizon Cloud on Microsoft Azure deployment needs.

Microsoft does not typically allow increasing quota levels in a free type of account. Therefore, a free account cannot be made to align with the requirements to support the Horizon Cloud deployment.

A PoC deployment might consider taking the following approach:

1. Sign up for the free Azure account that provides use of $200 Azure credits for 30 days from sign up.
2. Immediately convert that free Azure account to a pay-as-you-go account. The $200 Azure credits become available in that pay-as-you-go account for 30 days.
3. Sign up for the Horizon Universal Subscription License 60-day trial (a requirement if you do not already have a Horizon Cloud tenant).
4. Continue completing the Azure preparation items 2 - 6 during the time that VMware is configuring the Horizon Cloud tenant.
5. When the Welcome to Horizon Cloud email is received, log in and run the Add Pod wizard.

This way, the Azure subscription is already prepared and ready at the same time the Horizon Cloud tenant account is ready to log in and run the Add Pod wizard.

When you have obtained an Azure subscription in which you can perform the remaining five preparation activities, then you can log in to the Azure portal and begin those preparations.

The remaining Microsoft Azure preparation activities (2 - 5) all take place using the Azure portal, within your Azure subscription. Log in to the Azure Portal using the credentials for your subscription.

Register Essential Resource Providers

Now let's register all the essential resource providers that will be needed for the PoC pod deployment.

Before the next PoC activity of confirming the availability of required items in a specific Azure regional location, the Microsoft.DBforPostgreSQL, Microsoft.Sql, and Microsoft.Compute resource providers must be in Registered status, to make the Azure Portal display the right data.
By registering now all of the additional resource providers that the Add Pod wizard needs, it saves time for later. The ones needed by the Add Pod wizard will already be registered when you start running the Add Pod wizard.

In the Azure Portal, it can take up to 10 minutes for each resource provider to move from Unregistered to Registered status.

**Steps**

2. Into the portal’s upper search bar, start typing subscriptions to see a Subscriptions icon. Click that Subscriptions icon.

When you click Subscriptions, the portal displays the Subscriptions pane, and lists those subscriptions that are associated with your login credentials.
If you do not see the name of the subscription that you obtained to use for this PoC, click on the **Subscriptions == global filter**. Then in the subsequent box that appears, clear the **Show only subscriptions selected in** box and click **Apply** so that the filter says **Subscriptions == all**.

3. Click the subscription you want to use for this PoC.

4. On the subscription's pane, scroll down to locate **Resource providers**.
5 Click on that **Resource providers**, which opens the Resource providers pane.

6 For each one of the essential resource providers in the following table, scroll through the Resource providers pane and check if **Registered** is displayed next to that resource provider.

This screenshot depicts where to see the **Registered** status.
You will likely see some resource providers already display Registered status in a brand new Azure subscription because of Microsoft Azure standard behavior. For example, a new Azure subscription usually has Microsoft.MarketplaceOrdering in Registered status already, because Azure assumes that anyone with an Azure subscription would want to use the Azure Marketplace.

If anything along the lines of NotRegistered is displayed for one of these essential resource providers, then select that one and click the Register button at the top of the pane to move that one into Registered status.

After you click Register, the pane will display Registering, as in the following example screenshot.

Please note that the portal's Resource providers pane does not automatically refresh when the registering process is done. You must click Refresh to see the up-to-the-moment status. For each resource provider, it might take up to 10 minutes for its status to change from Registering to Registered.
Repeat the steps of checking and registering for the resource providers in the following table until they all display Registered status in the Resource providers pane for the subscription.

### Table 6-1. Essential Resource Providers for PoC

<table>
<thead>
<tr>
<th>Resource Provider</th>
</tr>
</thead>
<tbody>
<tr>
<td>Microsoft.Authorization</td>
</tr>
<tr>
<td>Microsoft.Compute</td>
</tr>
<tr>
<td>Microsoft.DBforPostgreSQL</td>
</tr>
<tr>
<td>microsoft.insights</td>
</tr>
<tr>
<td>Microsoft.KeyVault</td>
</tr>
<tr>
<td>Microsoft.MarketplaceOrdering</td>
</tr>
<tr>
<td>Microsoft.Network</td>
</tr>
<tr>
<td>Microsoft.ResourceGraph</td>
</tr>
<tr>
<td>Microsoft.ResourceHealth</td>
</tr>
<tr>
<td>Microsoft.Resources</td>
</tr>
<tr>
<td>Microsoft.Security</td>
</tr>
<tr>
<td>Microsoft.Sql</td>
</tr>
<tr>
<td>Microsoft.Storage</td>
</tr>
</tbody>
</table>

3 Check Availability and Quota Limits, Increase as Needed

In a Horizon Cloud on Microsoft Azure deployment, you decide on the specific Azure regional location in which to situate the deployment.

For low latency, one typically situates a Horizon Cloud on Microsoft Azure deployment in an Azure location that is geographically closest to the intended VDI end users.

However, because Microsoft can restrict specific Azure services and quota in a specific regional location at any point in time, it is important to have a short list of candidate locations that you will consider using for your PoC deployment.

As an example, see the following screenshot taken on the day when we checked the availability of Standard Dv3 Family vCPUs for France Central in our pay-as-you-go subscription. This screenshot depicts how Microsoft Azure did not have this key VM family available in that region for our subscription.
Best Practice Recipe

1. For each candidate region on your short list, verify availability of the Horizon Cloud on Microsoft Azure deployment’s required Azure Database for PostgreSQL Service and specific VM families.

2. When you see that one of those regions meets the availability of both the PostgreSQL database and the VM families, make that your region for this PoC deployment,

3. Increase that region’s VM Family vCPUs and Total Regional vCPUs enough to accommodate both the initial pod and day-2 items of adding a gateway, creating a few golden images, desktop pools, and multi-session farms.

Table 6-2. For that recipe, verify that your candidate location in your subscription allows creation of these items

<table>
<thead>
<tr>
<th>Item</th>
<th>Used for</th>
</tr>
</thead>
<tbody>
<tr>
<td>Azure Database for PostgreSQL - Generation 5, memory optimized, 2 vCores, 10 GB storage.</td>
<td>Pod itself</td>
</tr>
<tr>
<td>Standard Dv3 Family vCPUs - 10 vCPUs</td>
<td>8 vCPUs for the pod’s management VMs, plus 2 vCPUs for one RDS golden image (images added post-deployment)</td>
</tr>
<tr>
<td>Standard DSv2 Family vCPUs - 4 vCPUs</td>
<td>2 vCPUs for one single-session Windows 10 golden image and 2 vCPUs for one Windows 10 Enterprise multi-session golden image. (These images are created using the system’s automated Import VM from Marketplace wizard.)</td>
</tr>
<tr>
<td>Standard Av2 Family vCPUs - 9 vCPUs</td>
<td>The pod’s external gateway configuration (gateway added post-deployment) requires 8 vCPUs. Then for our PoC recipe, we are going to use a 1 vCPU VM from this Av2 family for our Active Directory domain and domain controller machine. That estimate calculates to 9 vCPUs (8 plus 1).</td>
</tr>
</tbody>
</table>
Table 6-2. For that recipe, verify that your candidate location in your subscription allows creation of these items (continued)

<table>
<thead>
<tr>
<th>Item</th>
<th>Used for</th>
</tr>
</thead>
<tbody>
<tr>
<td>Optional: <strong>Standard NVSv3 Family vCPUs</strong> - 12 vCPUs * (1+ number of desktops)</td>
<td>If you want to try a GPU-enabled golden image and desktops in your PoC. From this NVSv3 family, 12 vCPUs for the golden image, plus additional 12 vCPUs times the number of desktops you want to try based on that image.</td>
</tr>
<tr>
<td>VM families for PoC single-session virtual desktops and multi-session farms</td>
<td>The virtual desktops and remote apps served by the pod. A Horizon Cloud on Microsoft Azure deployment supports using a variety of VM families for these. A minimum of 2 vCPUs is recommended for each single-session or multi-session virtual instance. For a PoC recipe, we estimate using the <strong>Standard Dv3 Family vCPUs</strong> and 20 single-session Windows desktops, 2 multi-session Windows desktops, and 2 multi-session RDSH servers. That estimate calculates to 48 vCPUs from that family (24 x 2 vCPUs).</td>
</tr>
</tbody>
</table>

Please note that the above numbers reflect only those for a simple PoC recipe, as described in this page’s introduction. These numbers cannot be construed to accommodate complex pod deployments nor desktops or remote apps at scale nor upgrades of the initial deployment nor the service's Windows 11 OS support.

**Example of Checking Availability and Quota Limits**

First check if the Azure Portal prevents you from creating an Azure Database for PostgreSQL server - Single Server in your first selected candidate location. Then check the availability of the required **VM Family vCPUs** in that candidate.

Before performing these steps, ensure that PoC activity two is completed to have the Microsoft.DBforPostgreSQL, Microsoft.Sql, and Microsoft.Compute resource providers in **Registered** status.

The screenshots in this example illustrate what we saw for our pay-as-you-go subscription used to prove out this page's steps. Your display will be different, because Microsoft personalizes what you see and have access to.

**Step 1 - Initiate creation of the Azure Database for PostgreSQL in the location to identify the top contender.**

If Microsoft Azure prevents you from creating a single-server type of Azure Database for PostgreSQL instance in a specific regional location, that will block pod deployment also. So it’s best to prove this database requirement first.

1. Into the Azure Portal’s upper search bar, start typing **Azure Database for PostgreSQL servers** to see a **Azure Database for PostgreSQL servers** icon. Click that icon.
2 On the portal’s Azure Database for PostgreSQL servers pane, click Create. This step initiates the wizard process in which we can check if Microsoft Azure will allow creation in our candidate location.

3 In the Single server option, click Create. The pod deployer uses the Single server type and for verifying availability in a location, we need to compare like with like.

Even if the Azure Portal prompts you to create Flexible server, choose the Create Single Server path.

4 In the Single server pane, scroll to the Location menu and select your candidate regional location.

If the Azure Portal displays a message saying the service is not available in this location for your subscription, then try the next candidate on your short list of locations.

For example, on the day we did these steps in our pay-as-you-go subscription and selected (Asia Pacific) Southeast Asia, the message Currently, the service is not available in this location for your subscription. appeared.
Microsoft has total control over which locations it makes its services available, on a region-by-region basis and on a subscription-by-subscription basis.

On the same day, when we selected our next candidate of (Asia Pacific) East Asia, no message displayed.

When you see no message displayed under the Location menu about the selected location, that location is a viable candidate to move on to the next verification, to verify the VM families in that candidate location.

5 Close the Single server pane by clicking the X to close the pane. Allow the portal to discard unsaved edits.

Step 2 - Using the location identified in Step 1, check the availability of the VM Family vCPUs in that location.

1 Into the Azure Portal’s upper search bar, start typing quota to see a Quotas icon. Click that Quotas icon.

When you click Quotas, the portal displays the Quotas pane.
2 Click **Microsoft.Compute**.

The **My quotas** pane appears with its filtering boxes at the top, with the **Provider** filter set to **Microsoft.Compute**.

3 Select your candidate location in the **Location** menu and check that the **Subscription** menu has selected the subscription you are using for this PoC.

This screenshot illustrates selecting the location **West US 3** and our PoC subscription.
4 For your candidate location, check the **VM Family vCPUs** levels of availability for each of the following families, and increase the quota of that family if needed.

**Table 6-3. Family vCPUs for Pod Deployment and Post-Deployment VDI**

<table>
<thead>
<tr>
<th>VM Family</th>
<th>Available vCPUs Needed</th>
</tr>
</thead>
<tbody>
<tr>
<td>Standard Dv3 Family vCPUs</td>
<td>10 vCPUs total (8 vCPUs for the PoC pod itself, plus 2 vCPUs to use for post-deployment creation of one RDSH golden image post-deployment)</td>
</tr>
<tr>
<td>Standard DSv2 Family vCPUs</td>
<td>4 vCPUs total (for post-deployment creation of one single-session Windows 10 golden image and one Windows 10 Enterprise multi-session golden image)</td>
</tr>
<tr>
<td>Standard Av2 Family vCPUs</td>
<td>9 vCPUs total (8 vCPUs for a gateway on the pod and 1 vCPU for our PoC local Active Directory)</td>
</tr>
<tr>
<td>Optional: Standard NVSv3 Family vCPUs</td>
<td>12 vCPUs for the golden image plus 12 x number of desktops you intend to try out</td>
</tr>
<tr>
<td>Your desired VM families for PoC single-session VDI desktops and multi-session farms</td>
<td>For our PoC recipe, we planned on using the Standard Dv3 Family vCPUs and have 20 single-session Windows desktops, 2 multi-session Windows desktops, and 2 multi-session RDSH servers. For us, that calculates to 48 vCPUs from that Standard Dv3 Family vCPUs (24 x 2 vCPUs).</td>
</tr>
</tbody>
</table>

5 If for any of the VM families in step 3, you see the ⬅️ symbol (circle lowercase I) next to the VM family name, click that symbol. If you see a **VM size currently unavailable** message, you will have to strike that candidate from your list. If that happens, repeat the Step 1 - PostgreSQL database verification to identify a new viable candidate, and then repeat this check on the VM families.

For example, see the following screenshot taken on the day when we checked the availability of Standard Dv3 Family vCPUs for France Central in our pay-as-you-go subscription. This screenshot depicts how Microsoft Azure did not have this VM family available in that region for our subscription.
Your display will have different details, because Microsoft personalizes what you see and have access to.

6 When a family has less capacity available than the numbers in the table above, increase that family's quota in that region.

For example, this screenshot illustrates that our subscription has zero percent of **Standard Dv3 Family vCPUs** currently in use in the **West US 3** location (**0% Usage**). However, this screenshot also indicates our current quota accommodates usage only to a maximum of 10 (ten), which is too low a number. Because our PoC will require usage of more than 10 from **Standard Dv3 Family vCPUs**, we will need to increase that quota maximum number.

Microsoft provides multiple ways of requesting an increase in individual **VM Family vCPU** quotas. In our pay-as-you-go subscription, in the screen depicted above, we clicked the pencil icon to the right of the usage numbers.

Clicking that pencil icon opens a form to specify a request to increase the quota to a new maximum number of vCPUs for that VM family within your selected location and subscription.

**Note** Microsoft itself decides whether your request will be approved or rejected. If your request is rejected, a link will display where you can open a support request to Microsoft so that they can assist you with the increase.

7 After using the above steps to identify a location that satisfies the availability requirements for both the single-server Azure Database for PostgreSQL and the **VM Family vCPUs**, then check that regional location's **Total Regional vCPUs** level to see how many vCPUs are available, unused.
For example, this screenshot points out that for the **West US 3** location in our subscription, the **Total Regional vCPUs** total quota is a maximum of 10, which is far less than the 71 that our PoC recipe requires.

If you see that in your subscription the region’s **Total Regional vCPUs** quota level indicates that there are not enough unused vCPUs out of the total usage to fulfill the total available vCPUs needed for the PoC deployment, then you must also increase the **Total Regional vCPUs** level.

**Step 3 - Verify the Total Regional vCPUs in the candidate region, and increase if needed.**

Do these steps after increasing the quota limits for the individual **VM Family vCPUs**.

For the purposes of our PoC recipe, we want to accommodate at least 71 new total vCPUs in our desired Azure regional location. In Azure, the **Total Regional vCPUs** is the quota to check. (This 71 count includes the 16 vCPUs for the pod itself and one external gateway plus our local Active Directory domain server plus an estimated three golden images, and approximately 20 virtual desktops. This number does not cover use of GPU-enabled NV family images or desktops. To include those would add an additional 12 vCPUs plus 12 times the number of GPU desktops.)

1. In the same **My quotas pane** as in the preceding steps, select the location and subscription you will use for the deployment and find the row for the **Total Regional vCPUs**.

   As an example, the following screenshot depicts **Total Regional vCPUs** for the **West US 3** location and the subscription we will use for our deployment.

2. When the **X of Y** number displayed in the **Usage** column indicates that the number of available (unused) vCPUs is less than what is needed for the PoC, click the pencil icon at the right of the **X of Y** number to increase the maximum number (increase the Y).
Your specific numbers will look different from the ones in our screenshot, because your numbers will reflect what is up-to-the-minute in your own subscription and locations.

Look at the delta \((Y - X = Z)\) to verify how many vCPUs are still remaining available in the location for use. For example, if the Usage displays \(10\ of\ 15\), the delta would be only 5 available \((15 - 10 = 5)\). That low number would need to be increased to accommodate the PoC.

For our brand new pay-as-you-go subscription, we have no VMs in our subscription yet, and so our initial usage shows \(0\ of\ 10\) and we need to set our maximum to 71 to accommodate our estimate of having the pod, its external gateway, our Active Directory domain machine, three golden images, and 20 virtual desktops.

After clicking the pencil icon in our subscription, in Request quota increase, we enter 71 for the new maximum limit and submit the request.

**Note** Microsoft itself decides whether your request will be approved or rejected. If your request is rejected, a link will display where you can open a support request to Microsoft so that they can assist you with the increase.

**Specific example of checking the Standard Dv3 Family vCPUs availability in West US 3 location and increasing its quota in the subscription**

For our PoC recipe, we plan on using the **Standard Dv3 Family vCPUs** family for our single-session virtual Windows desktops, multi-session Windows desktops, and multi-session RDSH servers. To accommodate those plus the pod itself, we needed to check that family’s quota to have at least 58 vCPUs of that family (summed from the data in the previous table)

Checking for this number and increasing if needed will ensure we will not run short of **Standard Dv3 Family vCPUs** when we start to create virtual desktops.

1. In the My quotas pane, in the Search filter, type **Standard Dv3 Family** and set Location to West US 3. This search will reveal the available quotas for the **Standard Dv3 Family vCPUs** in West US 3 for our subscription.

2. Check the Usage column and confirm that the percent available has at least **58** remaining (unused) out of the total.
For example, if our Usage column says 8 of 10, that would mean that 8 out of the 10 are in use, and there are only 2 vCPUs remaining available in that quota level (10 minus 8 equals 2 unused). In that case, we must increase that quota by at least 56 vCPUs to accommodate the total 58 vCPUs available that we want for that Standard Dv3 Family vCPUs.

We then repeated a similar quota check for the other required VM families in the above table, and increased as needed.

4 Create App Registration

This app registration is a significant element in enabling a Horizon Cloud Service on Microsoft Azure deployment.

The app registration in your Azure subscription provides the ability for Horizon Cloud to use its API calls to create the Horizon Cloud Service on Microsoft Azure deployment in that subscription.

The service uses API calls to initially stand up the deployment in the subscription. The service also uses API calls for day-2 operations of creating golden images, VDI desktops, and so on — all of the VDI administration tasks.

Table 6-4. Collect these items during this procedure and save for when you run the Add Pod wizard

<table>
<thead>
<tr>
<th>Item for the Add Pod wizard</th>
<th>Your value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Subscription ID in Step 2 below</td>
<td>Your value</td>
</tr>
<tr>
<td>Application (client) ID in Step 5 below</td>
<td>Your value</td>
</tr>
<tr>
<td>Directory (tenant) ID in Step 5 below</td>
<td>Your value</td>
</tr>
<tr>
<td>Value of the client secret in Step 6 below</td>
<td>Your value</td>
</tr>
</tbody>
</table>

Steps

1. In the Azure Portal, navigate to the details for the subscription you are preparing to use in the PoC. Search for Subscriptions, and click on Subscriptions when you see it appear in the results list.

   For example, using the Azure Portal's search bar, search for Subscriptions, click on Subscriptions when you see it in the results list, then click on your specific subscription.

![Azure Portal showing search for Subscriptions](image)

2. From the subscription details, copy the Subscription ID and save it where you can retrieve it later for the Add Pod wizard.
The following screenshot shows where we copied the **Subscription ID** for our subscription named **Az POC for Horizon**. Our specific ID is redacted here to protect our values.

3. Then in the Azure Portal's search bar, search for **App registrations**, and click **App registrations** when you see it appear in the results list.

![Search for App registrations](image)

When you click **App registrations** from the search results, the portal displays the App registrations page.

4. On the App registrations page, click **New registration**.

![New registration](image)

Azure portal displays its UI for creating the app registration.

5. In the UI form, specify these items:
   - A display name that will remind you this registration is for Horizon Cloud use.
   - Select the single tenant choice for who can use this app registration (as of this writing, that choice is labeled **Accounts in this organizational directory only**).

6. Leave the optional items alone and click **Register**.

   The newly created app registration is displayed on screen.

7. From the displayed app registration, copy the **Application (client) ID** and **Directory (tenant) ID** and save them where you can retrieve them later for the Add Pod wizard.

   The following screenshot illustrates our app registration's essential details. Ours has display name **hcs-poc1**. Our specific **Application (client) ID** and **Directory (tenant) ID** are redacted here to protect our values.
8. Now create a client secret key for this app registration.

   a. From the app registration display in Step 5, click the text **Add a certificate or secret**.

      The portal displays the **Certificates & secrets** pane for that app registration.

      For our app registration named **hcs-poc1**, we saw:

      ![hcs-poc1 Certificates & secrets pane]

   b. Click **New client secret** in that pane.

      ![New client secret]

   c. The portal displays the **Add a client secret** screen.

      Type a description and select an expiration that aligns with a length of time to cover this Horizon Cloud on Microsoft Azure PoC.

      For us, we set a 12 months (one year) expiration. However, we will have to remember to return before it expires if we want to continue to use this same client secret with a new Horizon Cloud on Microsoft Azure deployment.

      We named our client secret **hcspoc1**.
Add a client secret

<table>
<thead>
<tr>
<th>Description</th>
<th>Value</th>
<th>Expires</th>
</tr>
</thead>
<tbody>
<tr>
<td>hspoc1</td>
<td></td>
<td>12 months</td>
</tr>
</tbody>
</table>

Add | Cancel

Click **Add**.

Immediately when you see the entry on the **Certificates & secrets** pane, locate the **Value** column, copy that value, and save where you can retrieve it later for the Add Pod wizard.

**Important** Keep this screen open until you copy the **Value** and save the value into a location where you can ensure you can retrieve it later. If you click away from this UI, the portal will obfuscate the **Value** and you will have to repeat creating a client secret to get a value to copy and save.

The following screenshot illustrates the client secret that we created. Our specific values are redacted here to protect our data.

![Client Secret Screenshot](image)

d  Click **Add**.

9  Now assign the Azure built-in **Contributor** role to this app registration.

This role assignment provides for Horizon Cloud’s ability to use its API calls for the PoC deployment in the subscription.

a  Again navigate back to the subscription details (try using the Azure Portal’s search bar, search for **Subscriptions**, click on **Subscriptions** when you see it, and then in the **Subscriptions** pane, click on the subscription).

b  Click **Access control (IAM)**.
c. In the **Access control (IAM)** pane, click **Add > Add role assignment**.

That action displays the **Add role assignment** pane.

d. In that **Add role assignment** pane, select **Contributor** and then click **Members** to move to the **Members** tab.
On the Members tab, keep User, group, or service principal selected and click Select members.

In the selection window, search for the name of the app registration that you created in Step 5.

When you created the app registration in Step 6, Azure also created an associated service principal of the same name as the app registration. Technically speaking, Horizon Cloud API calls use both the app registration and its associated service principal to create and work with the Horizon Cloud on Microsoft Azure deployment.

We search for the name we used for our app registration, hcs-poc1.
When you click the name, it becomes listed as selected member. Then click **Select** to finalize that selection.

The app registration's name is added to the **Members** tab. Add an optional description if you like, and then click **Review + assign** to move to the **Review + assign** tab.

This screenshot depicts our PoC choices, with the Object ID redacted.
On the **Review + assign** tab, complete these steps by clicking the button also labeled **Review + assign**.

1. **Member added and an optional description.**
2. **Click to proceed.**

---

5. **Set Up Networking**

Now let's create the virtual network (VNet) and subnets we'll use in our PoC.

For our PoC recipe, we will define the following address spaces and subnets.

Please note that Azure always reserves 5 addresses from every subnet for itself.
### Address Space

<table>
<thead>
<tr>
<th>Address Space</th>
<th>Name</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>10.0.0.0/23</td>
<td>vnet-hcspoc</td>
<td>The overall VNet that we’re creating for our PoC. The /23 starts the VNet’s address space with 512 addresses. This CIDR allows us to accommodate the pod, its gateways, and the golden images and virtual desktops for our PoC with the following subnets. Feel free to choose a larger space as you prefer for your PoC.</td>
</tr>
<tr>
<td>10.0.0.0/29</td>
<td>poc-adsubnet</td>
<td>We will place our local PoC Active Directory server machine on this subnet. We use /29 here because that’s the smallest range we can use and still accommodate the 5 addresses that Azure always reserves for itself from every subnet.</td>
</tr>
<tr>
<td>10.0.0.32/27</td>
<td>hcspoc-mgmt</td>
<td>For the Add Pod wizard’s pod management subnet. The Add Pod wizard enforces a minimum of /27 for this subnet. A Horizon Cloud on Microsoft Azure deployment requires that only the deployment’s VMs are to reside on this subnet, and no other pre-existing or post-deployment machines. Therefore, we must define this subnet and the following two subnets as distinct subnets. The Add Pod wizard also requires this subnet to have the service endpoint named Microsoft.Sql configured on it. We add that in the last step of this Activity.</td>
</tr>
<tr>
<td>10.0.0.64/28</td>
<td>hcspoc-uag-ext</td>
<td>For the Add Pod wizard’s external gateway subnet. The Add Pod wizard enforces a minimum of /28 for this subnet.</td>
</tr>
<tr>
<td>10.0.1.0/25</td>
<td>hcspoc-vdi</td>
<td>For the Add Pod wizard’s VM subnet. We use /25 here to provide for 128 addresses, which we use for the golden images and the VDI desktops that we planning for in our recipe here.</td>
</tr>
</tbody>
</table>

### Steps

1. In the Azure Portal, in the portal’s upper search bar, start typing **Virtual networks** to see a **Virtual networks** icon. Click that **Virtual networks** icon.
When you click **Virtual networks**, the portal displays the **Virtual networks** pane.

Verify the Subscription filter is set to your subscription for this PoC and click **Create**.

2 In the displayed Create virtual network wizard, use **Create new** to name and create a resource group to hold the VNet object in Azure.

In our case, we named this resource group **hcsvnet-RG**.

3 Type a name for your VNet and for the region, select the same Azure regional location that you verified using Activity, the one that meets the PoC's availability and quota needs.
For our PoC, we name our VNet as vnet-hcspoc and have chosen to use region West US 3 in our subscription, which is the region that we verified and increased quota from of Activity 3.

4 Move to the IP Addresses tab.

5 Azure pre-populates the IPv4 address space with a large value. Click on that pre-populated value and change the value to the CIDR you want to use for the VNet’s initial address space.
For our PoC, we choose to use **10.0.0.0/23**, a CIDR that provides for 512 IP addresses (10.0.0.0 – 10.0.1.255). As of this writing, when we click into the space below that value, the Azure Portal displays the address range.

Feel free to choose a larger address space here as you prefer for your PoC.

6. Now we specify the four subnets that we need for this PoC.

   For each of these subnets, you:
   
   a. Click **Add subnet**.

   Fill out the **Add subnet** UI, which at the time of this writing looks like the following screenshot. Type the subnet name and its address range and click **Add**.
With each click of Add, a subnet gets added to the IP Addresses tab. Repeat until all four subnets are listed on the IP Addresses tab.

<table>
<thead>
<tr>
<th>Subnet name</th>
<th>Subnet address range</th>
</tr>
</thead>
<tbody>
<tr>
<td>poc-adsubnet</td>
<td>10.0.0.0/29</td>
</tr>
<tr>
<td>hcs-poc-mgmt</td>
<td>10.0.0.32/27</td>
</tr>
</tbody>
</table>
### Subnet details

<table>
<thead>
<tr>
<th>Subnet name</th>
<th>Subnet address range</th>
</tr>
</thead>
<tbody>
<tr>
<td>hcspoc-uag-ext</td>
<td>10.0.0.64/28</td>
</tr>
<tr>
<td>hcspoc-vdi</td>
<td>10.0.1.0/25</td>
</tr>
</tbody>
</table>

7. Now the wizard has enough information to be submitted to create the VNet. Move to the **Review + create** tab.

    Azure runs its validation checks.
When you see a successful validation, click **Create**.

Azure begins deploying the VNet and subnets. When its deployment succeeds, the portal displays a notification similar to this one:

---

**Deployment succeeded**

Deployment `Microsoft.VirtualNetwork-20220216100004` to resource group `hcsvnet-RG` was successful.

- [Go to resource](#)
- [Pin to dashboard](#)

---
9. Now we need to add a service endpoint named `Microsoft.Sql` to our `hcspoc-mgmt` subnet that we created in the **Create virtual network** wizard.

a. In the portal, go to the newly created VNet (in our case, vnet-hcspoc).

b. Go to its list of subnets.

c. Click the `hcspoc-mgmt` subnet. The portal displays that subnet's details UI.

   We're going to add the service endpoint named `Microsoft.Sql`. 

   ![Image showing subnet details](image-url)
d. Click the **Service Endpoints** menu to get the list of services.
e Select Microsoft.Sql and then at the bottom of that UI, click Save.
Set Up a Local Active Directory VM on the VNet

Now we shall create a VM and configure it as a local Active Directory domain and domain controller to use with our PoC.
Why does our Horizon Cloud on Microsoft Azure PoC deployment need an Active Directory domain? Because:

- At core, VDI solutions are meant to provide virtual Windows desktops to end users.
- Historically, IT has used Microsoft Active Directory to hold the information about an organization's users and their IT-issued Windows computers (desktops).
- Therefore, having an Active Directory domain is a critical piece of a VDI solution like Horizon Cloud on Microsoft Azure.

By creating a local PoC Active Directory machine in our PoC's VNet, we will also have this machine provide the DNS (domain name services) that our PoC deployment needs for name resolution in the VNet.

Steps

1. In the Azure Portal, in the portal's upper search bar, start typing **Virtual machines** to see a **Virtual machines** icon. Click that **Virtual machines** icon.

   When you click **Virtual machines**, the portal displays the **Virtual machines** pane.

   ![Virtual machines pane](image)

   We verify that the Subscription filter is set to the subscription for our PoC and click **Create**.

   ![Create button](image)

2. Choose **Azure virtual machine**.
That action starts the **Create a virtual machine** wizard.

The following screenshot depicts what we saw at this point, at the time of this writing. As you can see from the side scroll bar, there are additional items further down in the wizard UI.

3 For our PoC Active Directory server, we select the following choices for the fields marked required (those with asterisks in the portal), and leave the optional ones at the defaults that the portal used.

- **Subscription** - We make sure this is set to the subscription for our PoC deployment.
- **Resource group** - We click **Create new** and type our chosen name **POC-AD**.
- **Virtual machine name** - We type **POC-AD**.
- **Region** - We select the same region as our PoC VNet (West-US3 for us).

  Below is a picture of our selections to this point. We must continue scrolling down to make choices for the next set of items.

![Create a virtual machine UI](image)

- **Image** - At the time of this writing, we can specify Generation 1 for our VM. Because this is a PoC and will be relatively short-lived, we want to go with a low generation VM which will allow us to choose a lower-cost VM size in the **Size** menu.

  First we click **Configure VM generation** to get the UI in which we can select **Generation 1** and apply our choice to the **Image** field.
After we apply **Generation 1**, we now click the **See all images** to navigate to the portal’s **Select an image** pane, locate the **Windows Server** tile there, and use that tile’s **Select** menu to look for **Windows Server 2019 Datacenter - Gen1**.

The following screenshot shows the list that we saw at the time of this writing.
From the list, we choose **Windows Server 2019 Datacenter - x64 Gen 1** for our PoC Active Directory server VM. Our reason for choosing this is because we have used this Windows Server 2019 Datacenter choice in the past for other situations and it seems good enough for our PoC purposes.

- **Size** - We choose **Standard_A1_v2**. As of this writing, Microsoft Azure has this available for its Gen 1 images for our subscription and region. One reason why we choose this VM size this is because this is a PoC and as of this writing, this VM size will cost us less per month than larger sizes. Another reason is because we’ve used this size before in other PoCs and it was fine in those other PoCs.

Below is an illustration of our selections for the preceding fields that the portal refers to as the instance details. Naturally we must continue scrolling down to make choices for the next set of items.

Please note that the displayed cost per month will vary depending on what Azure calculates for your subscription type, the selected region, and what Azure makes available for you.
Administrator account - We enter the information for what will be the admin account to log in to the server operating system when the VM is created. Follow the on-screen prompts. The Azure Portal will guide you on what conditions the admin name and password need to adhere to.

Inbound port rules - We select None. Later on, we will configure the use of Azure Bastion to enable us to log in to the VM’s system to configure the Active Directory domain.

Licensing - If you have an eligible Windows Server license, then you could select to use it. We do not have use of that for this PoC, so we left this unchecked.

Below is a picture of the fields we just completed, before moving to the next step.
Move on to **Next: Disks >**. On this Disks tab, we select **OS disk type** as **Standard HDD**. As of this writing, Standard HDD costs the least in Azure and because we are only using this machine for a PoC, we do not need a higher level disk.

Except for changing that **OS disk type**, we leave the other options as the defaults.
Move on to the **Next: Networking >**.

On the Networking tab we made the following selections, based on our PoC VNet and the specific subnet that we prepared for our PoC Active Directory in Activity 5:

- **Virtual network** - We select our `vnet-hcspoc`.
- **Subnet** - We select our `poc-adsubnet`.
- **Public IP** - We select None, because later we will use the Azure Bastion method of connecting to a VM. When using Azure Bastion, a public IP on the VM is unnecessary.
- **NIC network security group** - At the time of this writing, Azure had a default selected of **Basic**. We keep this for our PoC.
- **Public inbound ports** - At the time of this writing, Azure reflects the None choice we made earlier in this Create a virtual machine wizard. Therefore, we keep this setting.

- **Delete NIC when VM is deleted** - We choose this option. The reason why we select this is because this is a PoC for us and when we delete the VM at the end of our PoC, we want all of the VM’s artifacts also deleted at the same time.

Other that the preceding list, we made no additional choices on this Networking tab.

Below is a picture of the fields we just completed, before moving to the next step.

6 Now we click the Review + create button, because we are keeping the defaults on the remaining tabs and not making any new selections.
Azure runs its validation checks and, if validation passes, displays the final information for review. Use the scroll bar to review all the information of what will be created.

This screenshot depicts what was displayed for our situation and choices.

7 Then click **Create**.

Azure begins deploying the VM and all the related artifacts. When its deployment succeeds, the portal displays a notification similar to this one.
Now we need to log in to this new VM and configure it as our PoC Active Directory domain for our PoC Horizon Cloud on Microsoft Azure deployment.

a  In the portal, go to the newly created VM (in our case, POC-AD).

b  Ensure that the VM's **Agent status** shows **Ready**.

We will not be able to log in until this agent is ready. This agent is the Azure agent that Azure uses to manage the VM. Because the agent gets installed and runs in the VM's operating system, it can take several minutes for the agent to reach its ready state. You might have to use the **Refresh** button to refresh the values.

In this example, the agent is not ready yet.

In this example, the agent is ready and we can connect to the VM and log in.
Now we connect to the VM. We are going to use the Azure Bastion feature to connect to this VM and configure the features we need.

a. In the **Connect** menu, click **Bastion**.

![Bastion connection](image)

After clicking **Bastion**, the portal will display a screen for you to select deploying Bastion. This screenshot is an example, based on our PoC values for our VNet.
From this point, when we click the button **Deploy Bastion**, the Azure Bastion will be created in the listed VNet and resource, which is our PoC’s VNet and VNet's resource group.

In the Bastion deployment process, Azure will add a subnet for the Bastion to the VNet and create the Bastion in the indicated resource group.

b Click **Deploy Bastion**.

Azure adds the subnet for the Bastion, followed by creating the Bastion. The following screenshot illustrates the notification activity for our PoC when we took this step.

```
Notifications

More events in the activity log → Dismiss all

--- Creating a new Bastion
Creating a new Bastion 'vnet-hcspoc-bastion'.

a few seconds ago

Successfully added subnet
Successfully added subnet 'AzureBastionSubnet' to virtual network 'vnet-hcspoc'.

a few seconds ago
```

When the Azure Bastion is ready to use, the portal’s display refreshes to show the UI to log in to Bastion for the VM.

c Enter the admin credentials that you specified for the VM back in the Create a virtual machine wizard and click **Connect**.
Unless you clear the box about opening in a new window, Azure will start the connection in the same browser window. Our values are redacted here for privacy.

At this point, we are logged in to the VM's Windows Server 2019 operating system and presented with its standard desktop.

This screenshot is what we saw in our PoC at this point.
10 Now we configure this Windows Server 2019 as an Active Directory domain and domain controller for our PoC, and add in the admin accounts that the Horizon Cloud on Microsoft Azure PoC deployment needs.

**First we use the Add Roles and Features wizard to add the Active Directory Domain Services role and its required features.**

**Note** These steps are the same for configuring a Windows Server 2019 Datacenter to be an Active Directory domain and domain controller as you would find in many Internet articles and in the Microsoft documentation. Being a VM in the Azure cloud makes no difference in these steps.

a In that right hand blue **Networks** box about being discoverable, we choose **No**. For our PoC, we do not need this VM to be discoverable.

b In the Server Manager - Dashboard, from the upper right **Manage** menu, click **Add Roles and Features**.

The Add Roles and Features wizard is displayed.

c Proceed through the wizard making the selections to configure the server with the **Active Directory Domain Services** role and its required features.

- Select **Role-based or feature-based installation**.
- Select **Select a server from the server pool** and check that this step has the PoC VM selected. Ours is named **POC-AD**.
- Select the **Active Directory Domain Services** role.
- When the wizard displays a prompt about installing a list of role services or features that are also needed, use **Add Features** to include those also.
- When the wizard displays a step about installing additional features, keep the default selections and proceed to the next wizard step (**Next**).
- At the wizard's AD DS step, continue proceeding to the next confirmation step (**Next**).

The following screenshot depicts what we saw at our Confirmation step of the wizard. The left side shows the wizard steps that we went through and made our choices. In this screen, we deselect the box about restarting so that we can continue with our connection to the VM and watch the installation run.
■ Click **Install**.

The role installation activity starts running.

These screenshots depict what we saw.
d. Now promote the server to a domain controller. Click **Promote this server to a domain controller**.

**Now complete the steps to promote the server to a domain controller.**

After closing the Add Roles and Features wizard, the Active Directory Domain Services Configuration Wizard starts for getting the values to make this server a domain controller.

a. In the deployment configuration, select **Add a new forest** and then type in the root domain name you want for your PoC domain.

For our PoC, we use **hcspoc.local**.
b Proceed to the next wizard step for the Domain Controller Options.

Here, we keep the defaults as presented to us in the wizard for the forest and domain functional levels and we ensure that **Domain Name System (DNS) server** and **Global Catalog (GC)** are selected. (From the Microsoft documentation, Microsoft requires a Global Catalog for the first domain controller, and naturally this is our first one here.)

We also specify a DSRM password as the wizard prompts for it.
Proceeding to the next wizard step, a yellow message appears about how a delegation cannot be created. Click Show more to read the full message.

The reason for this message does not pertain to our PoC domain, because we know we have invented the domain name. So we ignore this yellow message and click Next to proceed.
d Verify the NetBIOS name that the wizard defaults to based on your typed domain name and change if you want.

For our PoC domain, we keep the **HCSPOC** name that the wizard derived from our entered hcspoc.local name.

![Image of Active Directory Domain Services Configuration Wizard]

e Proceed through the wizard.

For the Paths step, we kept the defaults.

On Review Options step, we reviewed that the wizard will configure this server as the first Active Directory domain controller in a new forest.

We also note that the Review Options states that this computer will be configured to use itself as its preferred DNS server. We decided that was fine for our PoC.
f Click **Next** to proceed to Prerequisites Check.

The following screenshot depicts what we saw. All prerequisite checks passed successfully. All of the yellow items are FYI for us, because they do not matter in our PoC.
Click **Install**.

When the system reaches the point where the machine needs to be restarted, the message **You're about to be signed out** is displayed. In the background, you can see the Results screen that the server was successfully configure.

Click **Close** on this message.
Then on the **You have been disconnected** message, click **Close** again to close the Bastion connection until the machine is back up and you see the Azure agent is ready.

![Disconnected](image)

**Disconnected**

---

**Reconnect to the VM and configure the admin accounts that our PoC needs**

Now that we have created our PoC Active Directory domain controller, we need to create three accounts that we can use in our PoC.

a. Reconnect to the VM’s operating system using **Connect > Bastion**.

When the Server Manager - Dashboard appears, we can see the dashboard reflects the just-configured **AD DS** and **DNS**.
b Now we need to add three user accounts to the domain.

These users accounts will be used in the steps in the section below Register the PoC Active Directory with the Horizon Cloud Tenant Register the PoC Active Directory with the Horizon Cloud Tenant.

For simplicity, because this is only a PoC, we will add all three of these accounts to the standard Domain Admins group in our PoC Active Directory.

For our PoC, we named our three accounts as:

- hcsbind1
- hcsbind2
- hcsjoin
To start adding the users to the Active Directory domain, from Server Manager - Dashboard, click **Tools > Active Directory Users and Computers**.

In that **Active Directory Users and Computers** tool, click **Action > New > User**.

Complete the fields for the first new user account. We named our first user **hcsbind1** and chose **User cannot change password** and **Password never expires**.

After you see that user listed, make it a member of the **Domain Admins** group.

Repeat step d to add two more users. We named ours **hcsbind2** and **hcsjoin**.

After the three user accounts are added, you can disconnect from the PoC Active Directory domain VM.
12. Now, because we have this VM as a DNS server, we must add its private IP address to the VNet's DNS Server configuration.

a. In the VM's overview details, make a note of the **Private IP Address**.

For our PoC VM, the address was **10.0.0.4**.

b. Now go to the VNet's settings and its DNS servers pane and click **Custom**.

c. Enter the private IP address from your PoC Active Directory VM and click **Save**.
Then go to your PoC Active Directory VM and restart it, as the on-screen message said to do.

Lastly, because we are finished needing to connect to the AD VM for the time being, we delete the Azure Bastion to prevent being charged for its hourly cost.

This is an optional step. If you don’t mind the hourly cost incurred for the Azure Bastion, it can be your own choice to keep it around. We decided to delete it for the savings.

Go to the resource group in which the Bastion was created and delete the bastion item.

This screenshot shows where the Azure Bastion existed in our VNet's resource group. We clicked into that Bastion and deleted it.
Deletion of the Azure Bastion can take around 10 minutes.

**Obtain Horizon Cloud Tenant Account**

Before you can log in and run the Add Pod wizard, you must have a cloud tenant account already set up and associated with your VMware Customer Connect account.

Prerequisites to getting a tenant account set up are:

- A VMware Customer Connect account or a VMware Cloud services account.
- A subscription that provides for access to the cloud-hosted services, such as the Horizon Universal Subscription. Refer to this [Horizon Subscription table](https://www.vmware.com/horizon-universal-subscription.html) a comparison of the various types.

**Getting an account**

Use the **Register** action in the header at [https://customerconnect.vmware.com](https://customerconnect.vmware.com).

**Obtaining the subscription**

If you know that you do not already have a subscription that provides access to the cloud-hosted services, one way to get a tenant account for your PoC is to sign up for a 60-day trial.

As of this writing, the known page about this 60-day trial license is this one: [https://www.vmware.com/horizon-universal-license-trial.html](https://www.vmware.com/horizon-universal-license-trial.html).

If you know that you already have a subscription or you know you are part of an enterprise account that has a subscription, then you might already have access to a cloud tenant account that is already set up.

To find out your current status, file a service request (SR) following the steps in VMware KB article [2006985](https://kb.vmware.com/s/article/2006985). You will need your current VMware Customer Connect account information.

**Notification when your tenant is set up**

When VMware associates your account credentials with the Horizon Cloud tenant account, an email is sent to the email address which is in the profile of that VMware Customer Connect account or VMware Cloud services account.

You will know that you have access to the tenant account when you see that email. Make sure to check your spam folder for emails from VMware Horizon Service.
Log In and Run the Add Pod Wizard

When you have the email stating that your tenant account is ready, you can log in and run the Add Pod wizard.

Make sure that your PoC Active Directory domain VM is running in the Azure Portal. After the previous steps, that VM is providing DNS services for the VNet which the Add Pod process will need.

Collect the following pieces of information and have them handy as you perform these steps. This information includes the pieces that you set up in the preceding activities.

Table 6-5. Collect these items during this procedure and use as you log in to the console and run the Add Pod wizard

<table>
<thead>
<tr>
<th>Item</th>
<th>Your Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Your VMware Customer Connect account, like <a href="mailto:name@example.com">name@example.com</a></td>
<td></td>
</tr>
<tr>
<td>Password for your VMware Customer Connect account</td>
<td></td>
</tr>
<tr>
<td>Subscription ID from Prepare Azure Activity</td>
<td>4</td>
</tr>
<tr>
<td>Directory (tenant) ID from Prepare Azure Activity</td>
<td>4</td>
</tr>
<tr>
<td>Application (client) ID from Prepare Azure Activity</td>
<td>4</td>
</tr>
<tr>
<td>Value of the client secret from Prepare Azure Activity Create App Registration. In the steps below, the Manage Subscriptions UI will refer to this as the Application Key.</td>
<td>4</td>
</tr>
<tr>
<td>VNet name from Activity</td>
<td>3</td>
</tr>
<tr>
<td>mgmt subnet name from Activity</td>
<td>3</td>
</tr>
<tr>
<td>vdi subnet name from Activity</td>
<td>3</td>
</tr>
</tbody>
</table>

1. Log in to Horizon Universal Console using your VMware Customer Connect account credentials.
   1. Log in to Horizon Universal Console using your VMware Customer Connect or VMware Cloud services credentials.
In a browser, go to https://cloud.horizon.vmware.com.

The login screen will automatically redirect to the VMware Cloud Services login UI at https://console.cloud.vmware.com.

Sign in using either your VMware Cloud services or your VMware Customer Connect credentials, following the on-screen prompts.

After you accept the terms of service, the main console displays the Getting Started page.
This page is the starting point for all new tenants.

Until you add a pod deployment, most of the console is locked to you.

So now we will start creating our PoC Horizon Cloud on Microsoft Azure deployment.

2. Add your Azure subscription information into the console.

As stated in Activity 4, this information is needed so that the service can use API calls to initially stand up the deployment in the subscription.

1. Click Manage > Manage Subscriptions.

Displays this UI window.
Keep the default **Add** action, because you’re adding in your subscription information for the first time, and type in a name you want used to refer to the subscription in the console. This name is solely to distinguish this subscription’s values from another one if you were going to use multiple Azure subscriptions with this Horizon Cloud tenant. We used myhcspoc.

For **Environment**, select **Azure - Commercial**.

Now copy into the four remaining fields the values that you collected in Activity 6.

**Note** Please note that this Manage Subscriptions UI’s **Application Key** field means the value of the client secret that you copied during Activity 2 Create App Registration.

**Table 6-6. The labels you’ll see in the Manage Subscriptions UI compared to the Azure Portal names from Activity 4**

<table>
<thead>
<tr>
<th>Manage Subscription UI</th>
<th>Name in Azure Portal</th>
<th>Your Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Subscription ID</td>
<td>Subscription ID</td>
<td></td>
</tr>
<tr>
<td>Directory ID</td>
<td>Directory (tenant) ID</td>
<td></td>
</tr>
<tr>
<td>Application ID</td>
<td>Application (client) ID</td>
<td></td>
</tr>
<tr>
<td>Application Key</td>
<td>Value of the client secret key</td>
<td></td>
</tr>
</tbody>
</table>

This screenshot shows our choices for our PoC. Our values are redacted for privacy.
3. When you have specified the required items, click **Confirm**.

The system starts to verify that all of the values are satisfactory, that they tie together as they are meant to from Activity.

When the system verification is successful that all of the values tie together satisfactorily, a blue success message briefly displays. We saw this message after we added our values.

At this point, you are back on the Getting Started page and can start the Add Pod wizard.

3. **Run the Add Pod wizard.**

1. Click **Manage > Add Pod**.

In this first wizard step, because you've already entered the subscription information, you can select your subscription name in **Apply Subscription**.
Add Microsoft Azure Capacity

1. Subscription
2. Pod Setup
3. Gateway Settings
4. Summary

Choose the Microsoft Azure subscription you want to apply or add a new one.

Pod Subscription

Apply Subscription: myhcspec
Subscription Name: myhcspec
Environment: AZURE
Subscription ID:
Directory ID:
Application ID:

Use a Different Subscription for External Gateway:

Click Next.

2. On the wizard's Pod Setup step, complete the two main areas — Details and Networking.

Details

- **Pod Name** - Type the name that you want for this pod when you see it in the console. (We used HCS-trialpod-1 for ours.)

- **Location** - Click Add, and then in the City Name field, start typing the name of a city.

  For ours, we started typing Arl to see Arlington names.
After a few letters, the system will start displaying names that match the letters you typed and click one of those cities that seems closest to what you want. (We chose Arlington, WA, United States.)

- **Microsoft Azure Region** - Select the same region in which you set all the quota, created the VNet, and the PoC Active Directory domain VM. (We are using West US 3.)

We left the rest of the items as-is in Details and then completed Networking.

**Networking**

- **Virtual Network** - Select your VNet. (Ours is vnet-hcspoc.)

- **Switch Use Existing Subnet to on position (green) and select the subnets that you created in that VNet.**

- **Management Subnet** - Select the mgmt subnet that you created in Activity 5. (Ours is hcspoc-mgmt.)

- **VM Subnet - Primary** - Select the vdi subnet that you created in Activity 5. (Ours is hcspoc-vdi.)

- **NTP Servers** - Enter a list of one of more NTP servers to use for time synchronization with the pod’s VMs. If you enter multiple names, separate them with commas. (We are using one named us.pool.ntp.org.)

The aforementioned items are the ones that we specifically set in this wizard step. The remainder we leave as-is, the defaults.

This screenshot illustrates our example.
Click **Next**.

3. On the wizard’s **Gateway Settings** step, switch **Enable External Gateway?** to **off** position.

Remember that in this PoC recipe's ingredients for this PoC section, we stated that this gateway can be added later after the pod is deployed. So we turn off these choices for now.
4 Click **Validate & Proceed**.

The system performs its validation checks based on what you entered into the wizard. When it all checks out, a blue message displays briefly.

![Pod and Gateway configurations validated successfully.](image)

The message text will mention the gateway even though we have switched off the gateway toggles for this deployment. This is to be expected.

5 In the wizard’s final **Summary** step, verify that it lists the correct **mgmt** and **vdi** subnets that you set up for your PoC.

Here is the full view from our example.
6 Click **Submit**.

The system starts generating the Horizon Cloud on Microsoft Azure deployment in your Azure subscription.

The console will reflect the progress state, starting with:
And then moving into:

The deployment can take between 30 to 45 minutes, depending on network traffic between Azure Cloud and Horizon Cloud.

The console will reflect when the process is completed.

When you see that Complete indicator, take the steps in the next section to register the PoC Active Directory domain with this Horizon Cloud on Microsoft Azure deployment.

**Register the PoC Active Directory with the Horizon Cloud Tenant**

Because this activity unlocks the rest of the console, you must complete it before you can start exploring your Horizon Cloud on Microsoft Azure deployment and start on day-2 tasks.

We set up the PoC Active Directory domain VM to make it easy to complete this activity.

1. When you see the deployment is successful on the console's Getting Started page, expand General Setup to see the Active Directory row.
   
   Click Configure in that row.
2. In the Register Active Directory window, enter the requested information about the PoC Active Directory domain and users created in Activity 6.

The required information is the PoC Active Directory domain's NetBIOS name, DNS domain name, and the short names and passwords for the AD users set up for this purpose.

Our PoC values are our NetBIOS name of **HCSPOC**, our DNS domain name **hcspoc.local**, and our two users named **hcsbind1** and **hcsbind2**. This screenshot depicts our entries.

Click **Domain Bind**.

The system saves the information and then displays the Domain Join window.

3. In the Domain Join window, enter the PoC Active Directory VM's IP address and the credentials for the third AD user created in Activity 6.
Our PoC values are our VM’s 10.0.0.4 IP address and the credentials for our user named **hcsjoin**. This screenshot depicts our entries.

![Domain Join screenshot](image)

Click **Save**.

The system saves the information and then displays the Add Administrator window.

4. In the Add Administrator window, start typing the letters for **Domain Admins** until you see the system’s search has located the **Domain Admins** group in your PoC Active Directory domain.

![Add Administrator screenshot](image)

Click **Domain Admins** to select that AD group. That is the AD group that three user accounts created in Activity 6 were added to as members.
Then click Save.

5. At this point, the system automatically and immediately logs you out of the console. You will see something similar to this:

```
Welcome to
VMware Horizon®

Logout complete

RETURN TO LOGIN PAGE
```

Please note that this forced logout is by design.

Now that the Active Directory domain is registered with the Horizon Cloud on Microsoft Azure deployment, authentication to your cloud tenant will have two gates: one for authenticating with the tenant account credentials and one for authenticating with an AD user account that is a member of the selected AD Domain Admins group.

6. Return to the main login page and log in again as you did before using your account credentials. From cloud.horizon.vmware.com, the system will automatically redirect to the VMware Cloud Services login UI to complete the login flow.
After logging in, an Active Directory Credentials window displays. The NetBIOS name from Step 2 above is displayed in the window.

Log in using the credentials of one of the user accounts that are members of the AD Domain Admins group.

For our PoC, we use the credentials of our `hcsjoin` account, as illustrated below.
After completing those two authentication gates, you are back in the Horizon Universal Console — and now all of the left-hand navigation areas are accessible!

Usually at this point, a window **What's New in Horizon Cloud** window pops up. This window can be closed - you can easily display it from the upper help menu. The help menu looks like a circled ?, as depicted here:
Start Using Horizon Cloud

Congratulations! By reaching this point, you have successfully completed this step-by-step recipe for a pod to use in a simplified, kick-the-tires exploration of Horizon Cloud Service on Microsoft Azure deployment.

Now you can begin your exploration and our recipe ends here.

Final Notes

One of the simplifications we chose at the beginning of this recipe was to first deploy without an external Unified Access Gateway configuration and then add that later.

To perform a number of the day-2 PoC activities, you will want to add that external Unified Access Gateway configuration.

A prerequisite to running the Edit Pod wizard to add an external Unified Access Gateway configuration is you must provide a signed SSL certificate that meets specific criteria.
The reason why this signed SSL certificate is needed is the Unified Access Gateway capabilities require SSL for client connections. The certificate must be signed by a trusted Certificate Authority (CA). The signed SSL server certificate must be in PEM format and based on an FQDN. 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.

For reference, see the following pages.

- Convert an SSL Certificate to the Required PEM Format
- Add a Gateway Configuration to a Horizon Cloud On Microsoft Azure Deployment

Online Videos and Additional Content from VMware Tech Zone

VMware Digital Workspace Tech Zone provides a Horizon Cloud on Microsoft Azure evaluation guide. This guide includes videos that give a visual understanding of the deployment process:
Evaluation Guide for VMware Horizon Cloud Service on Microsoft Azure

Horizon Cloud on Microsoft Azure - First Pod Deployment - High-Level Workflow

This workflow describes the high-level sequence starting with your first use of the Horizon Universal Console to deploy a pod-manager-based pod to using the Horizon Control Plane services to configure virtual desktops and apps. The sequence ends with your end users launching their entitled virtual desktops and apps.

**Note** This workflow is specifically about a pod based on the pod-manager technology for Horizon Cloud on Microsoft Azure, and not about deploying a Horizon pod using Horizon Connection Server technology.

**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 Console Used 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.

1. Fulfill the prerequisites. See VMware Horizon Cloud Service on Microsoft Azure Requirements Checklist For New Pod Deployments.
2 Perform the preparatory tasks outside of Horizon Cloud. See *Preparing to Deploy a Horizon Cloud Pod into Microsoft Azure*.

3 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 Log in to the Horizon Universal Console and run the wizard to deploy the pod.

5 Register your Active Directory domain with the Horizon Control Plane, which includes providing the names of service accounts. Ensure these service accounts meet the requirements described in *Service Accounts That Horizon Cloud Requires for its Operations*.

6 Assign the appropriate roles to individuals in your organization for authenticating to and performing operations in the administrative console. There are two types of roles that are used in Horizon Cloud. See *Best Practices about the Two Types of Roles You Give to People to Use the Cloud-Based Console to Work in Your Horizon Cloud Environment*.

7 Complete the Universal Broker configuration for your tenant. See *Configure the Universal Broker Settings*.

8 Create the required CNAME records in your DNS server. For information about the purpose of these CNAMEs and CNAME record requirements for Universal Broker, see *How to Obtain the Horizon Cloud Pod Gateway’s Load Balancer Information to Map in Your DNS Server* and *Configure the Universal Broker Settings*.

**Note** When you have the external gateway configuration using a private IP address for the configuration’s Azure load balancer, then you must have a firewall or NAT managing Internet traffic to that private IP address, and that firewall or NAT must provide a public IP address and be configured such that the FQDN specified when deploying the external gateway configuration is publicly resolvable. The control plane must be able to communicate with the FQDN specified for the external gateway.

9 Optional: If your Horizon Cloud tenant account is not already onboarded with the VMware Cloud services engagement platform, consider onboarding it now. Onboarding to the platform is a prerequisite to activating the Horizon Infrastructure Monitoring features for your pods deployed in Microsoft Azure. See *Onboard Your Horizon Cloud Tenant to VMware Cloud Services Engagement Platform Using the Cloud-Based Console*.

10 Optional: Activate Horizon Infrastructure Monitoring. See *Horizon Infrastructure Monitoring and the Pods in Your Horizon Cloud Environment*.

11 Create a golden image. Creating a golden image is a multi-step process. For a high-level overview of the various ways of creating a golden image that can be used in your Horizon Cloud tenant, see *Creating Desktop Images for a Horizon Cloud Pod in Microsoft Azure*. Creating a golden image starts with importing a base VM, which you then customize for your business and end-user needs.
Depending on the type of end-user assignment for which the image is ultimately intended, perform one or more of the following steps as appropriate.

- In a single-session image that will be used for provisioning single-session 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 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 Microsoft Windows Client Operating System, Install the Appropriate GPU Drivers, and Five Key Steps to Take with Your Golden Images to Get Optimal Remote Experience Performance.

  **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 multi-session image that will be used for provisioning multi-session desktops and remote applications, install the third-party applications you want to provide to your end users from that multi-session image and configure other applicable customizations, such as setting desktop wallpaper, installing the 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 or Windows 11 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 Microsoft Windows Server Operating System, Customize the Image VM's Microsoft Windows 10 Enterprise Multi-Session Operating System, Install the Appropriate GPU Drivers, and Five Key Steps to Take with Your Golden Images to Get Optimal Remote Experience Performance.

  **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.

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.

To provision session desktops and remote applications from a published multi-session image:

a Create a desktops 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.
b Create an applications 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.

15 To provision single-session VDI desktops from a published single-session 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 section about creating these desktop assignments.

16 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 entitle those end users to the base desktops in which they can use those applications. The application assignment entitles use of the user’s entitled App Volumes applications within the Windows operating system of the user’s entitled desktop. See App Volumes Applications - Overview and Prerequisites.

17 When your deployment has a two-factor authentication configuration, you must complete the following tasks:

- If the pod’s external gateway has two-factor authentication configured and the two-factor authentication server is not reachable within the same VNet topology into which the gateway's Unified Access Gateway instances are deployed, configure that two-factor authentication server to allow communication from the IP address of the external gateway's load balancer.

  In this scenario where the two-factor authentication server is not reachable within the same VNet topology as the gateway deployment, the Unified Access Gateway instances attempt contact with that server using that load balancer address. To allow that communication traffic, ensure the load balancer resource’s IP address that is in that external gateway's resource group is specified as a client or a registered agent in your two-factor authentication server's configuration. Refer to the documentation for your two-factor authentication server for the specifics on how to allow that communication.

- If your two-factor authentication server is reachable within the same VNet topology, configure the two-factor authentication server to allow communication from the appropriate NICs that were created for the deployment’s Unified Access Gateway instances in Microsoft Azure.

  Your network administrator determines the two-factor authentication server’s network visibility to the Azure VNet topology and its subnets used for the deployment. The two-factor authentication server must allow communication from the IP addresses of the Unified Access Gateway instances’ NICs that correspond to the subnet for which your network administrator has given network visibility to the two-factor authentication 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 and its gateways go through an update.
To support communication traffic between the gateway and the two-factor authentication server both for ongoing pod operations and after each pod update, ensure the IP addresses of those four NICs are specified as clients or as registered agents in that server’s configuration. Refer to the documentation for your two-factor authentication server for the specifics on how to allow that communication.

For information on how to obtain those IP addresses, see Update Your Two-Factor Authentication System with the Required Horizon Cloud Pod Gateway Information.

After the above workflow steps are completed, give your Universal Broker brokering FQDN to your end users. They use that brokering FQDN in their Horizon Client or Horizon HTML Access (the web client) to launch their entitled desktops and remote applications.

You can find in-depth details on how to accomplish each workflow step in the topics that are linked from each step above.

Preparing to Deploy a Horizon Cloud Pod on Microsoft Azure

Before you log in to the Horizon Universal Console and run the pod deployment wizard for the first time, you must perform these preparatory tasks. This pod deployment wizard deploys the pod-manager-based type of pod.

Remember This wizard deploys the pod-manager-based type of pod into Microsoft Azure. The console currently does not provide a deployment wizard to automate a Horizon pod in Azure VMware Solution (AVS), which uses the Horizon Connection Server technology.

Fulfill the prerequisites described in the Chapter 3 VMware Horizon Cloud Service on Microsoft Azure Requirements Checklist For New Pod Deployments Starting From the v2207 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 the Chapter 3 VMware Horizon Cloud Service on Microsoft Azure Requirements Checklist For New Pod Deployments Starting From the v2207 Service Release:

- Ensure that your subscription does not restrict use of Azure StorageV2 account type. The App Volumes features require a storage account.

- If you are not planning to specify custom resource tags in the pod deployment wizard, ensure that your subscription does not require specific tag names on its resource groups.

- Ensure that your subscription does not have Azure Policies on it that would block, deny, or restrict creation of the pod's components in that subscription.

**Caution** If your subscription has restrictions involving creating resource groups, the pod deployment process can fail early on. If your subscription does not match those preceding items, the first step of creating the resource group for the pod manager VMs will fail to complete. Therefore, if your pod deployment process times out after an hour, 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, NVv4-series, and NCv2-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 When Your Organization Prefers to Use a Custom Role for the Horizon Cloud App Registration.

- 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 a Horizon Cloud App Registration in the Pod’s Subscription.

**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 required by Horizon Cloud to operate 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 When Your Organization Prefers to Use a Custom Role for the Horizon Cloud App Registration, 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 When Your Organization Prefers to Use a Custom Role for the Horizon Cloud App Registration 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, a 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.

Also, the role must be assigned directly to the service principal used for Horizon Cloud. The use of a group-based assignment of a role to the service principal — in which the role is assigned to a group and the service principal is a member in that group — is unsupported.

3 Verify that the resource providers that Horizon Cloud requires are all showing Registered status, as described in Resource Providers That Horizon Cloud Requires to Be in Registered Status in the Microsoft Azure Subscription.

4 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.

5 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.

6 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.

7 If you are not already registered for access to Horizon Cloud, verify that one of these two items is in place:

- Obtain a VMware Customer Connect account and register Horizon Cloud access with that account.
- If your group or organization used the VMware Cloud Services Engagement Platform to register for Horizon Cloud or they registered for use of Horizon Cloud using VMware Cloud services or Workspace ONE, you would have received an invitation email with an activation link when they invited you to join their VMware Cloud services org space in, as described in the VMware Cloud services documentation topic Add Users to Your Organization. If you received such an email, complete the email's instructions before trying to access Horizon Cloud.

After you have completed those preparatory tasks, sign in to the Horizon Universal Console at cloud.horizon.vmware.com. That address will redirect to the VMware Cloud Services login where you sign in using either your VMware Cloud Services credentials or VMware Customer Connect credentials.
After completing the login flow, you’ll see the console’s **Add Cloud Capacity** section and can click **Manage > Add Pod** to start the pod deployment wizard. Complete the wizard by entering the required information in each screen. For detailed steps, see Using the Horizon Universal Console to Perform an Automated Deployment of a Pod into Microsoft Azure.

**Note**  Login authentication into the cloud-based console relies on authenticating account credentials with VMware Cloud Services. If that service is unable to complete the necessary authentication requests, then 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.workspaceone.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. When you activate the Horizon Infrastructure Monitoring feature for a pod in Microsoft Azure, your pod’s subscription needs the appropriate quota and configuration to support deployment of the Horizon Edge Virtual Appliance into that subscription.

**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.

**Note**  GPU-enabled VMs are only available in some Microsoft Azure regions. See Microsoft Azure Products by region 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.
Pod Manager VMs

These VMs are generally considered the heart of the pod itself. The pod manager VMs are responsible for facilitating the connection of the end-user clients to the Horizon agent software running in the pod-provisioned virtual desktops.

Starting with the v2204 service release, new Horizon Cloud on Microsoft Azure deployments are deployed with high availability configured by default. The deployment has two pod manager VMs.

Table 6-7. Pod Management VM Requirements - For the Pod’s Core VMs, Not Including Gateway Configurations

<table>
<thead>
<tr>
<th>VM</th>
<th>Microsoft Azure VM Specification</th>
<th>Quantity</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pod manager instances</td>
<td>Linux - Standard Dv3 Family:</td>
<td>2 per pod during steady-state operations</td>
<td>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 switched over to using the new green components. See Horizon Cloud Pods - Maintenance and Updates for a description of the pod’s blue/green update process.</td>
</tr>
<tr>
<td></td>
<td>Standard_D4_v3 (4 cores, 16 GB memory). OS disk: Standard HDD 30 GiB</td>
<td>4 per pod during the end-to-end time for the pod's blue/green update process.</td>
<td>Note: 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>
</tr>
</tbody>
</table>

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.
<table>
<thead>
<tr>
<th>VM</th>
<th>Microsoft Azure VM Specification</th>
<th>Quantity 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 completed and the pod is switched over to... |
### Table 6-8. 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></td>
<td></td>
<td>using the new green components. See Horizon Cloud Pods - Maintenance and Updates for a description of the pod's blue/green update process.</td>
</tr>
</tbody>
</table>

### Table 6-9. 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 Connect or 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 Images - General

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.

Golden images are either GPU-enabled or not, depending on your selection when you create them.

In Horizon Cloud, you can create both single-pod golden images and multi-pod golden images. Creation of either type is done using the console's automated Import VM from Marketplace wizard.

The automated wizard automatically uses a specific VM size by default. This default is based on its internal settings and on your choices in the wizard for the specific operating system (OS) and whether to include GPU.

### Golden Image VMs

As of the v2207 service release, single-pod images and multi-pod images both have the same VM model requirements. Single-pod images are imported using the console's Imported VMs page. Multi-pod images are imported using the console’s Multi-Pod Images page.
Table 6-10. Golden Image 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>Golden images</td>
<td>For GPU-enabled golden images,</td>
<td>Varied,</td>
<td>A golden image is a Microsoft Windows operating system VM that is based on your needs.</td>
</tr>
<tr>
<td></td>
<td>the system uses:</td>
<td>based on your</td>
<td>A Windows single-session operating system VM provides the base used to create the VDI desktops.</td>
</tr>
<tr>
<td></td>
<td>■ The Standard_NV12s_v3</td>
<td>needs</td>
<td>An RDS-capable Windows operating system VM provides the base used to create the VMs in farms that provide session-based desktops and remote</td>
</tr>
<tr>
<td></td>
<td>(from the Standard NVSv3</td>
<td></td>
<td>applications to your end users. This RDS-capable category includes both Windows Server OSes and Windows Enterprise multi-session OSes.</td>
</tr>
<tr>
<td></td>
<td>Family vCPUs)</td>
<td></td>
<td>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</td>
</tr>
<tr>
<td></td>
<td>■ OS disk: Standard HDD 127</td>
<td></td>
<td>to provide:</td>
</tr>
<tr>
<td></td>
<td>GiB</td>
<td></td>
<td>■ RDSH desktops using Microsoft Windows 2016 Datacenter, no GPU</td>
</tr>
<tr>
<td></td>
<td>For non-GPU-enabled golden</td>
<td></td>
<td>■ RDSH desktops using Microsoft Windows 2016 Datacenter, with GPU</td>
</tr>
<tr>
<td></td>
<td>images using non-Windows 11</td>
<td></td>
<td>Then you need at least 2 golden image VMs.</td>
</tr>
<tr>
<td></td>
<td>OSes, the system uses:</td>
<td></td>
<td>The process of converting a golden image into a published image is sometimes called publishing the image, or also called sealing the image.</td>
</tr>
<tr>
<td></td>
<td>■ The Standard_DS2_v2</td>
<td></td>
<td>The resulting published image is sometimes called a sealed image or an assignable image, because it is in a finalized state for use in</td>
</tr>
<tr>
<td></td>
<td>(from the Standard DSv2 Family</td>
<td></td>
<td>assignments.</td>
</tr>
<tr>
<td></td>
<td>vCPUs)</td>
<td></td>
<td>The system automatically powers off the golden image at completion the publishing workflow. When you update a published image, the system</td>
</tr>
<tr>
<td></td>
<td>■ OS disk: Standard HDD 127</td>
<td></td>
<td>powers the VM on again.</td>
</tr>
<tr>
<td></td>
<td>GiB</td>
<td></td>
<td><strong>Note</strong> When you duplicate a single-pod image using the console’s Duplicate action, the system temporarily powers on the golden image’s VM</td>
</tr>
<tr>
<td></td>
<td>For non-GPU-enabled golden</td>
<td></td>
<td>to obtain its configuration for the duplicate, and then powers it off again.</td>
</tr>
<tr>
<td></td>
<td>images with Microsoft Windows 11</td>
<td></td>
<td>For information about how to create a single-pod golden image, see the topic Creating Desktop Images and Your Horizon Cloud Pods.</td>
</tr>
<tr>
<td></td>
<td>OS or a Windows 11 OS</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Enterprise multi-session OS, the</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>system uses:</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>■ The Standard_D4s_v3</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>(from the Standard DSv3 Family</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>vCPUs)</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>■ OS disk: Standard HDD 127</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>GiB</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**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 6-11. 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>.</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>

**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).
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 6-12. 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>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 6-12. 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>
<tr>
<td></td>
<td><strong>Note</strong> 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).</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

### Horizon Edge Virtual Appliance

This appliance is created in a pod's Microsoft Azure subscription when you use the Horizon Universal Console to activate Horizon Infrastructure Monitoring on that pod.

### Table 6-13. Horizon Edge Virtual Appliance Requirements

<table>
<thead>
<tr>
<th>VM</th>
<th>Microsoft Azure VM Specification</th>
<th>Quantity</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Horizon Edge Virtual Appliance</td>
<td>Linux - Standard Dv3 Family: Standard_D4_v3 (4 cores, 16 GB memory). OS disk: Standard HDD 30 GiB</td>
<td>1 per pod</td>
<td>When the Horizon Infrastructure Monitoring feature is activated on a pod, this VM is created and used to collect the monitoring data and communicate that data to the Cloud Monitoring Service (CMS).</td>
</tr>
</tbody>
</table>

### Special Case Support-Related Jump Box VM

If you make a support request to VMware and the support team determines the way to service that request is to deploy a temporary jump box VM for communication with the VMware-managed appliances, your subscription cores and quota will need to accommodate that deployment at that time. Permission will be requested from you for a support-related jump box deployment.
This jump box would be deployed under supervision of the VMware Support team and deleted under supervision of the VMware Support team when the VM is no longer needed in service of your support request.

Table 6-14. Temporary Support-Related 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>Support-related jump box</td>
<td>Linux Standard F Family:</td>
<td>1</td>
<td>This support-related jump box VM is designed for secure communications to the VMware-managed appliances in service of your support request to VMware Support.</td>
</tr>
<tr>
<td></td>
<td>Standard_F2 (2 cores, 4 GB memory)</td>
<td></td>
<td>OS disk: Standard HDD 30 GiB</td>
</tr>
</tbody>
</table>

**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, when you deploy an external gateway into its own subscription, that deployment also requires 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 reuse existing subnets that already have items that are using IP addresses on those subnets. If an IP address is already in use on the subnets, issues such as the pod failing to deploy and other downstream IP-conflict issues have a high likelihood of occurring. Do not put any resources on these subnets or otherwise use any of the IP addresses. This caution notice includes pods deployed from Horizon Cloud — do not reuse subnets on which you already have a deployed pod.
<table>
<thead>
<tr>
<th>Into which VNet are you deploying the external gateway?</th>
<th>Subnet creation</th>
<th>Subnets needed</th>
</tr>
</thead>
</table>
| When deploying a pod with the external gateway using the pod's VNet | For this configuration, you can either create subnets in advance on the VNet and specify those subnets 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. **Important** 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 In Advance of Pod Deployment, Create the Horizon Cloud Pod's Required Subnets on your VNet in Microsoft Azure. 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. | Pod deployment using this configuration requires following subnets:  
- Management subnet, for IP addresses used by the VMs involved in management activities of the pod itself.  
- Primary VM subnet — 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... |
<table>
<thead>
<tr>
<th>Into which VNet are you deploying the external gateway?</th>
<th>Subnet creation</th>
<th>Subnets needed</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>in the deployment wizard, the Unified Access Gateway VMs also consume IP addresses from this subnet.</td>
</tr>
<tr>
<td></td>
<td></td>
<td><strong>Important</strong> 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.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>DMZ subnet, for IP addresses used by the optional external Unified Access Gateway configuration.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>When you have the deployer automatically create the subnets, the deployer always creates the new subnets in the corresponding VNet.</td>
</tr>
<tr>
<td>Into which VNet are you deploying the external gateway?</td>
<td>Subnet creation</td>
<td>Subnets needed</td>
</tr>
<tr>
<td>------------------------------------------------------</td>
<td>----------------</td>
<td>---------------</td>
</tr>
<tr>
<td></td>
<td></td>
<td>In terms of the VNet's address space, the deployer handles the subnet address spaces you enter into the wizard as follows:</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- If you specify address spaces in the wizard that are not already in the VNet's address space, the deployer automatically updates the VNet's configuration to add those address spaces. Then it creates the new subnets in the VNet.</td>
</tr>
<tr>
<td></td>
<td></td>
<td>- If the address spaces specified in the wizard are already contained within the VNet's existing address space, the deployer creates the new subnets in the VNet using the specified address spaces.</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 subnets 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

- Management subnet, for IP addresses used by the VMs involved in management activities of the external gateway itself (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

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 is intended for use 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 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?

Subnet creation

address spaces for the subnets that the pod deployer will automatically create or specify subnets that you have created in advance.

Subnets needed

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

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

**Procedure**

1. In the Microsoft Azure portal, navigate to the **Virtual Networks** pane by using the portal’s search bar to search for virtual networks and selecting the corresponding virtual networks result.

2. In the **Virtual Networks** pane, start the VNet creation wizard by clicking **Create**.

3. In the wizard, specify the following information in the on-screen wizard steps.

<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 select 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.

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 and Related Service Features and Horizon Cloud Pod - Ports and Protocols Requirements.

Important  The pod manager VMs 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 gateway connector VM having outbound Internet access. If you require proxy-based outbound Internet access, you must 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 collocated 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.
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 reuse existing subnets that already have items that are using IP addresses on those subnets. If an IP address is already in use on the subnets, issues such as the pod failing to deploy and other downstream IP-conflict issues have a high likelihood of occurring. Do not put any resources on these subnets or otherwise use any of the IP addresses. This caution notice includes pods deployed from Horizon Cloud — do not reuse subnets on which you already have a deployed pod.

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.

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 and Related Service Features, 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 V Nets 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 and Related Service Features.

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 V Nets must be peered and you and your networking team must ensure that your peered V Nets 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 and Related Service Features.

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 and Related Service Features and Horizon Cloud Pod - Ports and Protocols Requirements.

DNS Requirements for a Horizon Cloud Pod in Microsoft and Related Service Features

For successful end-to-end use of the service's features, you must ensure the Domain Name Service (DNS) names described in this documentation article are resolvable and reachable from
the management and tenant subnets using the specific ports and protocols identified in this article's tables.

The pod deployment process requires that the VMs have access to those hosts (DNS names) over your selected VNet. After the pod is successfully deployed in your subscription, various day-to-day service operations require network access to specific DNS names, as does the pod update process for updating the pod's software when new software is made available. This article describes these DNS requirements.

**Some Overarching Key Points**

**About these required DNS names**

Deploying and running a Horizon Cloud pod requires network access to specific DNS addresses over your Microsoft Azure VNets. For the pod deployer to work, you must configure your firewalls to allow the network access to those addresses. The purpose of each DNS address is stated in the following tables.

In addition to allowing network communication to those DNS addresses, the DNS configuration on your VNet must resolve the names as described in this article.

When you select the option to deploy the external gateway in its own VNet separate from the pod manager's VNet, that VNet's subnet must meet the same DNS requirements as the pod manager's VNet's management subnet.

Besides the pod deployer and its workflows, various service features require access to specific DNS addresses for those features to work end to end. Those DNS names are also provided in the following tables.

Some of these DNS names have a regional element to them.

The activation of Horizon Infrastructure Monitoring on a pod will automatically instantiate the Horizon Edge Virtual Appliance in the pod manager's VNet and management subnet. The Horizon Edge Virtual Appliance has its own DNS name requirements. Therefore, before you activate Horizon Infrastructure Monitoring on a pod, ensure you meet the DNS requirements of the Horizon Edge Virtual Appliance as indicated in the following tables.

As described in **Tight Integration Within the VMware Ecosystem**, you can use Horizon Cloud with other products available from the broader VMware ecosystem. Those other products might have additional DNS requirements. Such additional DNS requirements are not detailed here. For such DNS requirements, see the documentation set for the specific products that you will be integrating with your pod.

**About ports and protocols after a pod is deployed, for ongoing service-related operations**

After a pod is successfully deployed, specific ports and protocols are required for ongoing Horizon Cloud operations. See **Horizon Cloud Pod - Ports and Protocols Requirements** for details.
Regional Control Plane DNS Names

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 6-15. 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 Of Europe</td>
<td>cloud-eu-central-1.horizon.vmware.com</td>
</tr>
<tr>
<td>AP_SOUTHEAST_2 Of Australia</td>
<td>cloud-ap-southeast-2.horizon.vmware.com</td>
</tr>
<tr>
<td>PRODI_NORTCENTRALUS2_CP1 Of USA-2</td>
<td>cloud-us-2.horizon.vmware.com</td>
</tr>
<tr>
<td>PRODI_NORTHEUROPE_CP1 Of Europe-2</td>
<td>cloud-eu-2.horizon.vmware.com</td>
</tr>
<tr>
<td>PRODI AUSTRALIAEAST_CP1 Of 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>UK</td>
<td>cloud-uk.horizon.vmware.com</td>
</tr>
<tr>
<td>Europe-3</td>
<td>cloud-de.horizon.vmware.com</td>
</tr>
</tbody>
</table>

For the regional DNS addresses listed in the following tables for the Horizon Edge Virtual Appliance, those addresses use the same regions as the regional control plane DNS names.

DNS Requirements for the Overarching Pod Deployment Process, Pod Updates, Activation of Various Service Features, and Ongoing Operations

For successful end-to-end use of the service's features, 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 tables. Some of the service features that require the ability to reach specific DNS names are:

- The pod deployer that automatically deploys a pod-manager-based Horizon pod into your Microsoft Azure subscription
- The pod update feature, that updates the pod’s software to a more recent software version
- The import-image process that uses the Import from Marketplace wizard
- The agent-related features, such as automated agent update (AAU)
- Universal Broker
- Horizon Infrastructure Monitoring and its Horizon Edge Virtual Appliance
Features related to the Cloud Management Service (CMS)

Most especially for pod deployment, pod updates, and when activating the Horizon Infrastructure Monitoring on a pod

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 tables. The appliances used in these workflows use specific outbound ports to securely download the software that these processes need into your Microsoft Azure environment. Those DNS names are also used so that the appropriate workflow-related appliances can communicate with the cloud control plane.

For new pod deployments, you must configure your network firewall, network security group (NSG) rules, and proxy servers such that the key deployment-related appliances have the ability to contact the DNS addresses on the ports that they require. Otherwise, the pod deployment process will fail.

Activating Horizon Infrastructure Monitoring on a pod will instantiate the Horizon Edge Virtual Appliance into the same VNet and management subnet as that pod’s pod-manager appliances. You must ensure the configuration your network firewall, NSG rules, and proxy servers allows the Horizon Edge Virtual Appliance to contact the DNS addresses on the ports that it requires. Otherwise, deployment of that appliance will fail.

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.
DNS Requirements for New Pod Deployment, Pod Updates, and Service Operations that Apply on a Tenant-Wide Basis

The following table describes the DNS requirements for a new pod deployment, pod updates, and service operations that are applicable on a tenant-wide basis. For Horizon Infrastructure Monitoring activation and Horizon Edge Virtual Appliance DNS requirements, see the section after this one. Because the Horizon Infrastructure Monitoring feature is activated on a per-pod basis, its DNS requirements warrant their own descriptive table.

**Attention** Starting in early 2021, as a result of an upgrade to the service's regional control plane instances, the d1mes20qfad06k.cloudfront.net DNS name is no longer required for any of the regional control plane instances. All of the regional control plane instances now use the hydra-softwarelib-cdn.azureedge.net DNS name. The following table's contents is aligned with that current reality.
<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 control plane instance is</td>
<td>443</td>
<td>TCP</td>
<td>Regional control plane instance</td>
</tr>
<tr>
<td></td>
<td>specified in your Horizon Cloud tenant account. The regional instance is set when the</td>
<td></td>
<td></td>
<td>- United States: cloud.horizon.vmware.com, cloud-us-2.horizon.vmware.com</td>
</tr>
<tr>
<td></td>
<td>account is created, as described in Deployments and Onboarding to Horizon Cloud for</td>
<td></td>
<td></td>
<td>- Europe: cloud-eu-central-1.horizon.vmware.com, cloud-eu-2.horizon.v</td>
</tr>
<tr>
<td></td>
<td>Microsoft Azure and Horizon Pods.</td>
<td></td>
<td></td>
<td>mware.com</td>
</tr>
<tr>
<td></td>
<td>- cloud-us-2.horizon.vmware.com</td>
<td></td>
<td></td>
<td>mware.com</td>
</tr>
<tr>
<td></td>
<td>- cloud-eu-central-1.horizon.vmware.com</td>
<td></td>
<td></td>
<td>- Japan: cloud-jp.horizon.vmware.com</td>
</tr>
<tr>
<td></td>
<td>- cloud-eu-2.horizon.vmware.com</td>
<td></td>
<td></td>
<td>- United Kingdom: cloud-uk.horizon.vmware.com</td>
</tr>
<tr>
<td></td>
<td>- cloud-ap-southeast-2.horizon.vmware.com</td>
<td></td>
<td></td>
<td>- Germany: cloud-de.horizon.vmware.com</td>
</tr>
<tr>
<td></td>
<td>- cloud-ap-2.horizon.vmware.com</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>- cloud-jp.horizon.vmware.com</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>- cloud-uk.horizon.vmware.com</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>- cloud-de.horizon.vmware.com</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Management</td>
<td>softwareupdate.vmware.com</td>
<td>443</td>
<td>TCP</td>
<td>VMware software package server. Used for downloading updates of the</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>agent-related software used in the system’s image-related operations.</td>
</tr>
</tbody>
</table>
Table 6-16. DNS Requirements for New Pod Deployment, Pod Updates, and Service Operations that Apply on a Tenant-Wide Basis (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>Management</td>
<td>hydra-softwarelib-cdn.azureedge.net</td>
<td>443</td>
<td>TCP</td>
<td>Horizon Cloud content delivery server. On the management subnet, this site is used by the service for downloading necessary binaries used by the pod infrastructure.</td>
</tr>
<tr>
<td>Management</td>
<td>packages.microsoft.com</td>
<td>443 and 11371</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>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>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>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>Management</td>
<td>esm.ubuntu.com</td>
<td>80 and 443</td>
<td>TCP</td>
<td>Ubuntu software package server. Used by the pod’s Linux-based VMs for tracking security updates for high and critical CVEs (Common Vulnerabilities and Exposures) in the Ubuntu base OS and scale-out infrastructure.</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. |
Table 6-16. DNS Requirements for New Pod Deployment, Pod Updates, and Service Operations that Apply on a Tenant-Wide Basis (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): management.azure.com  
  - Microsoft Azure Germany: management.microsoftazure.de  
  - Microsoft Azure China: management.chinacloudapi.cn  
| 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. |
Table 6-16. DNS Requirements for New Pod Deployment, Pod Updates, and Service Operations that Apply on a Tenant-Wide Basis (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:  
  1. Global Azure SQL service tag: Sql  
  2. 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 host name 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. |
Table 6-16. DNS Requirements for New Pod Deployment, Pod Updates, and Service Operations that Apply on a Tenant-Wide Basis (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 names, depending on which regional control plane instance is specified in your Horizon Cloud tenant account. The regional instance is set when the account is created, as described in Deployments and Onboarding to Horizon Cloud for Microsoft Azure and Horizon Pods.  
- connector-azure-us.vmwarehorizon.com  
- connector-azure-eu.vmwarehorizon.com  
- connector-azure-aus.vmwarehorizon.com  
- connector-azure-jp.vmwarehorizon.com  
- connector-azure-uk.vmwarehorizon.com  
- connector-azure-de.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-uk.vmwarehorizon.com  
- United Kingdom: connector-azure-de.vmwarehorizon.com |
| Management    | Depending on which regional control plane applies to your Horizon Cloud account:  
- North America: kinesis.us-east-1.amazonaws.com  
- Europe, Germany: kinesis.eu-central-1.amazonaws.com  
- Australia: kinesis.ap-southeast-2.amazonaws.com  
- Japan: kinesis.ap-northeast-1.amazonaws.com  
- United Kingdom: kinesis.eu-west-2.amazonaws.com | 443  | TCP      | Cloud Monitoring Service (CMS) |
Table 6-16. DNS Requirements for New Pod Deployment, Pod Updates, and Service Operations that Apply on a Tenant-Wide Basis (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 - pod manager VMs, Unified Access Gateway VMs, gateway connector VM, Horizon Edge Virtual Appliance | If your firewall or network security group (NSG) supports the use of service tags, apply Azure service tag AzureCloud. If your firewall or NSG does not support the use of service tags, use the host name:  
- Commercial environments - Allow host name monitor.horizon.vmware.com  
- US Federal environments - Please open a case with VMware Federal Support to request the monitoring system host name. | 1514 1515 | TCP | Used for system monitoring |
| Tenant | hydra-softwarelib-cdn.azureedge.net | 443 | TCP | Horizon Cloud content delivery server. On the tenant subnet, this site is used by various system image-related processes, including processes involved in the system's automated Import Image from Marketplace workflow and agent pairing workflow. |

Horizon Edge Virtual Appliance DNS Requirements

Activating Horizon Infrastructure Monitoring on a pod will instantiate the Linux-based Horizon Edge Virtual Appliance in your Microsoft Azure subscription. The Horizon Edge Virtual Appliance is deployed into the pod's subscription with a NIC on the pod's management subnet — the same management subnet as that pod's pod-manager appliances. In the following table, the listed purposes are in context of this virtual appliance.

Note There are no DNS names with uk in this table like there are in the preceding table. As stated in the Release Notes, this Horizon Infrastructure Monitoring feature is currently in Limited Availability.
### Table 6-17. Horizon Infrastructure Monitoring DNS Requirements

<table>
<thead>
<tr>
<th>Subnet Source</th>
<th>Destination (DNS name)</th>
<th>Port/Protocol</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Management</td>
<td>azure.archive.ubuntu.com</td>
<td>80 and 443 / TCP</td>
<td>Ubuntu software package server. Used by the Linux-based appliance for Ubuntu operating system updates.</td>
</tr>
<tr>
<td>Management</td>
<td>archive.ubuntu.com</td>
<td>80 and 443 / TCP</td>
<td>Ubuntu software package server. Used by the Linux-based appliance for Ubuntu operating system updates.</td>
</tr>
<tr>
<td>Management</td>
<td>changelogs.ubuntu.com</td>
<td>80 and 443 / TCP</td>
<td>Ubuntu software package server. Used by the Linux-based appliance for Ubuntu operating system updates.</td>
</tr>
<tr>
<td>Management</td>
<td>security.ubuntu.com</td>
<td>80 and 443 / TCP</td>
<td>Ubuntu software package server. Used by the Linux-based appliance for Ubuntu operating system updates.</td>
</tr>
<tr>
<td>Management</td>
<td>esm.ubuntu.com</td>
<td>80 and 443 / TCP</td>
<td>Ubuntu software package server. Used by the Linux-based appliance for tracking security updates for high and critical CVEs (Common Vulnerabilities and Exposures) in the Ubuntu base OS and scale-out infrastructure.</td>
</tr>
<tr>
<td>Management</td>
<td>*.blob.core.windows.net</td>
<td>443 / TCP</td>
<td>Used for programmatic access to the Azure Blob Storage. Used to download the Docker images from those DNS addresses that the appliance’s module requires.</td>
</tr>
<tr>
<td>Management</td>
<td>*.blob.core.windows.net, or one of the region-specific names below, depending on which regional control plane applies to your tenant account: North America: edgehubprodna.azure-devices.net</td>
<td>443 / TCP</td>
<td>Used to connect the appliance to the Horizon Cloud control plane, to download configurations for the appliance’s module, and to update the appliance’s module’s runtime status.</td>
</tr>
<tr>
<td>Management</td>
<td>europe: edgehubprodeu.azure-devices.net</td>
<td>443 / TCP</td>
<td>Used to connect the appliance to the Horizon Cloud control plane, to download configurations for the appliance’s module, and to update the appliance’s module’s runtime status.</td>
</tr>
<tr>
<td>Management</td>
<td>australia: edgehubprodap.azure-devices.net</td>
<td>443 / TCP</td>
<td>Used to connect the appliance to the Horizon Cloud control plane, to download configurations for the appliance’s module, and to update the appliance’s module’s runtime status.</td>
</tr>
<tr>
<td>Management</td>
<td>japan: edgehubprodjp.azure-devices.net</td>
<td>443 / TCP</td>
<td>Used to connect the appliance to the Horizon Cloud control plane, to download configurations for the appliance’s module, and to update the appliance’s module’s runtime status.</td>
</tr>
<tr>
<td>Management</td>
<td>*.docker.com, *.docker.io, docker.io, cloud.google.com, gcr.io, ghcr.io, dl.k8s.io, k8s.gcr.io, storage.googleapis.com, cloud.weave.works, packages.cloud.google.com, apt.kubernetes.io</td>
<td>443 / TCP</td>
<td>Used to download Docker binaries and Kubernetes binaries that the appliance needs.</td>
</tr>
</tbody>
</table>
### Table 6-17. Horizon Infrastructure Monitoring DNS Requirements (continued)

<table>
<thead>
<tr>
<th>Subnet Source</th>
<th>Destination (DNS name)</th>
<th>Port/Protocol</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Management</td>
<td>If your firewall or network security group (NSG) supports the use of service tags, apply Azure service tag AzureCloud. If your firewall or NSG does not support the use of service tags, use the host name: Commercial environments - Allow host name monitor.horizon.vmware.com US Federal environments - Please open a case with VMware Federal Support to request the monitoring system host name.</td>
<td>1514 and 1515 / TCP</td>
<td>Used for system monitoring</td>
</tr>
</tbody>
</table>

### If Required for An Active Support Request, Temporary Jump Box Ports and Protocols

If you make a support request to VMware and the support team determines the way to service that request is to deploy a temporary jump box VM for SSH communication with the VMware-managed appliances, that jump box requires the ports and protocols described here.

Permission will be requested from you for a support-related jump box deployment. The VMware support team will inform you of any requirements, as appropriate for any support situation.

This support-related jump box VM is designed to communicate as a source to the following destinations:

- The pod's pod manager VMs' port 22 using SSH and port 22.
- The Unified Access Gateway VMs' port 9443 using HTTPS.
The gateway connector VM’s port 22 using SSH, in a deployment where the external gateway is deployed in its own VNet.

The Horizon Edge Virtual Appliance port 22 using SSH, in a deployment with Horizon Infrastructure Monitoring configured.

Because these VMs are assigned IP addresses dynamically, the following network rules can provide for the described communications. During the support-request activities, always seek guidance and supervision from VMware Support for requirements for the support-related jump box deployment.

- 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.

### Horizon Cloud Pod - Ports and Protocols Requirements

This page is a reference for all of the possible ports and protocols used for communication within a typical Horizon Cloud Service on Microsoft Azure deployment. Use these tables to ensure your network configuration and firewalls will allow the communication traffic that is required for a successful pod deployment and day-to-day operations.

The specific ports and protocols required for your particular deployment will in part depend on which features you select to use for your Horizon Cloud Service on Microsoft Azure deployment. If you plan to not use a specific component or protocol, then its required communication traffic is not necessary for your purposes, and you can ignore the ports associated with that component. As an example, if your end users will only use the Blast Extreme display protocol, then allowing the PCoIP ports is not a requirement.

**Important** In addition to the ports and protocols described here, pod deployment and its day-to-day operations have specific DNS requirements. For details, see DNS Requirements for a Horizon Cloud Pod in Microsoft and Related Service Features. For information about other ports supported for VMware products, go to VMware Ports and Protocols.

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 Default Network Security Group Rules for the VMs in a Horizon Cloud Pod.

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

In addition to the DNS requirements, the following tables list ports and protocols that are required for the pod to operate properly for ongoing operations after deployment. Some of these tables also list ports and protocols that are required for specific scenarios or when you have enabled particular features on the pod.
In the Microsoft Azure portal, the pod manager VMs have names that contain a part like `vmw-hcs-podID`, where `podID` is the pod’s UUID, and a node part.

**Note** Starting with the v2204 service release, new Horizon Cloud Service on Microsoft Azure deployments are deployed with high availability configured by default. The deployment has two pod manager VMs. In the tables below, wherever you see the phrase pod manager VM, that term applies to both pod manager VMs unless otherwise indicated.

The system’s use of a Microsoft Azure load balancer with the pod manager VMs started from manifest 1600, at the September 2019 service release. Therefore, all pods deployed new from manifest 1600 onward have a pod Microsoft Azure load balancer. Pods that were first deployed prior to manifest 1600 and subsequently updated to later manifests also have a pod Microsoft Azure load balancer. The table rows that mention the pod’s load balancer apply for all such pods.

### Table 6-18. 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>Pod manager VM</td>
<td>Pod’s other pod manager VM</td>
<td>4101</td>
<td>TCP</td>
<td>This traffic is JMS routing between the pod manager VMs.</td>
</tr>
<tr>
<td>Pod 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>Pod manager VM</td>
<td>808</td>
<td>HTTP</td>
<td>Health checks of the VMs in the load balancer’s backend pool. For pods deployed prior to v2204 release on which the high-availability toggle was not set and high availability not yet added, the load balancer has one pod manager VM in its backend pool to check.</td>
</tr>
<tr>
<td>Pod manager VM</td>
<td>Domain controller</td>
<td>389</td>
<td>TCP, UDP</td>
<td>Registering your Horizon Cloud tenant with an Active Directory is a requirement. The console’s Register AD Domain workflow must be performed after onboarding your first pod. This port is required for LDAP services when LDAP will be specified in that workflow. LDAP is the default for most tenants. Target is the server that contains a domain controller role in the Active Directory configuration.</td>
</tr>
<tr>
<td>Pod manager VM</td>
<td>Global catalog</td>
<td>326</td>
<td>TCP</td>
<td>Registering your Horizon Cloud tenant with an Active Directory is a requirement. The console’s Register AD Domain workflow must be performed after onboarding your first pod. This port is required for LDAP services when LDAP will be specified protocol in that workflow. LDAP is the default for most tenants. Target is the server that contains global catalog role in the Active Directory configuration.</td>
</tr>
</tbody>
</table>
Table 6-18. Pod Operations Ports and Protocols (continued)

<table>
<thead>
<tr>
<th>Source</th>
<th>Target</th>
<th>Ports</th>
<th>Protocol</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pod manager VM</td>
<td>Domain controller</td>
<td>88</td>
<td>TCP, UDP</td>
<td>Kerberos services. Target is the 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>Pod manager VM</td>
<td>Domain controller</td>
<td>636, 3269</td>
<td>TCP</td>
<td>Registering your Horizon Cloud tenant with an Active Directory is a requirement. The console’s Register AD Domain workflow must be performed after onboarding your first pod. These ports are required for LDAP over SSL (LDAPS) services only when LDAPS will be the specified protocol in the configuration for that registered AD. LDAPS can only be specified for the registered AD when your tenant is enabled to use the service’s LDAPS feature. Otherwise, LDAP is the requirement by default.</td>
</tr>
<tr>
<td>Pod manager VM</td>
<td>DNS server</td>
<td>53</td>
<td>TCP, UDP</td>
<td>DNS services.</td>
</tr>
<tr>
<td>Pod 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>Pod manager VM</td>
<td>True SSO Enrollment Server</td>
<td>32111</td>
<td>TCP</td>
<td>True SSO Enrollment Server. Required if you are using True SSO with your Horizon pods. 32111 is the default port used in the Enrollment Server installation. This port number can be configured during the Enrollment Server installation process per your requirements. See also the True SSO and Certificate Management and Horizon Cloud on Microsoft Azure deployments section in this topic.</td>
</tr>
<tr>
<td>Pod manager VM</td>
<td>Workspace ONE Access service</td>
<td>443</td>
<td>HTTPS</td>
<td>Note This row is applicable to environments with a single-pod-broker configuration. This information is not for environments with a Universal Broker configuration. In a single-pod-broker configured environment, the Workspace ONE Access Connector communicates with a pod to obtain the end-user entitlements (the assignments). Optional if you are not integrating Workspace ONE Access with the pod. In a single-pod-broker configured environment, this connection is used to create a trust relationship between the pod and the Workspace ONE Access service, where the Workspace ONE Access Connector is synched with the pod. Ensure that the pod can reach the Workspace ONE Access environment you are using on port 443. When 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>

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 6-19. Pod Operations Ports and Protocols**

<table>
<thead>
<tr>
<th>Source</th>
<th>Target</th>
<th>Port</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>Connector VM</td>
<td>NTP server</td>
<td>123</td>
<td>UDP</td>
<td>NTP services. Server that provides NTP time synchronization.</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 instances.

**Table 6-20. 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 manager VMs 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>
<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>
Table 6-20. 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>Your RSA SecurID Authentication Manager server</td>
<td>5555</td>
<td>TCP</td>
<td>When using RSA SecurID two-factor authentication with that Unified Access Gateway configuration. Shown here is the default value used for the communication port for RSA SecurID Authentication API agents for agent authentication.</td>
</tr>
<tr>
<td>Unified Access Gateway instances and their Microsoft Azure load balancer</td>
<td>Your syslog server</td>
<td>Port specified when you save the syslog server information to the gateway configuration.</td>
<td>TCP when TLS is selected in the configuration. UDP when UDP is selected in the configuration.</td>
<td>Optional. After a gateway configuration is already deployed, you can edit the configuration to specify your syslog server. The Unified Access Gateway instances send events to your syslog server through the instances’ load balancer. The port and protocol must match the selections you choose when you update the gateway configuration with your syslog server information.</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 6-21. 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>Rando...</td>
<td>Universal Broker service</td>
<td>443</td>
<td>HTTPS initially, then WebS...</td>
<td>Establishes a persistent WebSocket connection with the Universal Broker service.</td>
</tr>
</tbody>
</table>
Ports and Protocols Required by the Horizon Edge Virtual Appliance

When you activate Horizon Infrastructure Monitoring for a pod, the Horizon Edge Virtual Appliance is deployed and configured in that pod's subscription. The appliance has a single NIC on the pod's management subnet and an NSG is automatically configured on that NIC. That NSG’s rule specifies the traffic allowed into the appliance and out from the appliance. The following table lists the ports and protocols from that NSG that are needed during the activation process, as the appliance is deployed and configures the pod manager VMs so that it can collect the monitoring data it is designed to collect from those components. This table also lists the ports and protocols that are needed during the appliance’s steady-state operations of collecting the data it is designed to collect in this current service release.

Table 6-22. Port Requirements for the Horizon Edge Virtual Appliance

<table>
<thead>
<tr>
<th>Source</th>
<th>Target</th>
<th>Ports</th>
<th>Protocol</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Horizon Edge Virtual Appliance</td>
<td>DNS server</td>
<td>53</td>
<td>TCP</td>
<td>DNS services</td>
</tr>
<tr>
<td>Horizon Edge Virtual Appliance</td>
<td>Pod manager VMs</td>
<td>9172</td>
<td>Any</td>
<td>In steady-state operations over the management subnet, for the appliance to collect the monitoring data that it is designed to collect.</td>
</tr>
</tbody>
</table>
Table 6-22. Port Requirements for the Horizon Edge Virtual Appliance (continued)

<table>
<thead>
<tr>
<th>Source</th>
<th>Target</th>
<th>Ports</th>
<th>Protocol</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Horizon Edge Virtual Appliance</td>
<td>Cloud control plane</td>
<td>443</td>
<td>TCP</td>
<td>In steady-state operations over the management subnet, for the appliance to communicate the monitoring data to the cloud control plane. Also, during deployment of the appliance, this port is used to download specific externally located software components, such as the Microsoft Azure CLI software, that are required for the appliance to perform its configuration of the pod manager VMs for collection of the monitoring data.</td>
</tr>
<tr>
<td>Horizon Edge Virtual Appliance</td>
<td>Pod manager VMs</td>
<td>4002</td>
<td>TCP</td>
<td>In steady-state operations over the management subnet, for the appliance to collect the monitoring data that it is designed to collect.</td>
</tr>
</tbody>
</table>

End-User Connection Traffic Ports and Protocols Requirements

To connect to their pod-provisioned virtual desktops and remote applications from their devices, your end users use a compatible installed VMware Horizon Client or their browser (known as the Horizon HTML Access client). The ports which must be opened for traffic from the end users' clients 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 6-1. Illustration of the Horizon Cloud Pod Architecture where the Pod has 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 6-1. Illustration of the Horizon Cloud Pod Architecture where the Pod has 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 6-2. 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. 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 show 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 have no Unified Access Gateway configurations on the pod

**Note** For production environments where the tenant is configured to use single-pod brokering, the best practice for internal end-user connections is to use an internal Unified Access Gateway gateway configuration on the pod. Those connections would go to that gateway configuration in the single-pod brokering scenario.

In a configuration where you have single-pod brokering and Workspace ONE Access integrated with the pod, you typically have your end users connect through Workspace ONE Access. In this scenario, you must configure Workspace ONE Access and the Workspace ONE Access Connector pointing directly to the pod. 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 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 6-23. 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.</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 6-23. 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>8443 or 443, depending on what is set in your deployment</td>
<td>TCP</td>
<td>Blast Extreme via Blast Secure Gateway on Unified Access Gateway for data traffic from as Horizon Client. For a Horizon Cloud pod, this port is selected using the <strong>Blast Extreme TCP Port</strong> menu in the deployment wizard. Ensure that your networking allows the clients the outbound access to whichever one you specified for the external gateway. This URL is used by the clients to establish the Horizon Blast session to the Unified Access Gateway instances, through the load balancer that sits in front of those instances. Starting with the October 2021 service release, for new deployments of a gateway configuration, the deployer provides the ability select either 8443 or 443 for the Blast Extreme TCP port that the deployer will configure in the corresponding Unified Access Gateway configuration. Previously, the deployer configured 443 by default without providing a choice. If your gateway configuration was deployed prior to the date of the October 2021 service release, that configuration typically has 443 port set in the Blast External URL field in its Unified Access Gateway administration settings. <strong>Note</strong> Port 8443 is preferred because it is more efficient, provides better performance, and uses less resources on the Unified Access Gateway instances. Port 443, is less efficient and less performant. Using port 443 will result in CPU congestion in the instances. Port 443 would be used in your deployment only if your organization has set client-side restrictions, such as your organization only permits 443 outbound and not 8443.</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>8443</td>
<td>UDP</td>
<td>Blast Extreme via Blast Secure Gateway on Unified Access Gateway for data traffic (adaptive transport).</td>
</tr>
</tbody>
</table>
Table 6-23. 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>Browser</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.</td>
</tr>
<tr>
<td>Browser</td>
<td>Microsoft Azure load balancer for these Unified Access Gateway instances</td>
<td>8443 or 443, depending on what was set in your deployment</td>
<td>TCP</td>
<td>Blast Extreme via Blast Secure Gateway on Unified Access Gateway for data traffic from the Horizon HTML Access client (web client). For a Horizon Cloud pod, this port is selected using the Blast Extreme TCP Port menu in the deployment wizard. Ensure that your networking allows the clients the outbound access to whichever one you specified for the external gateway. This URL is used by the Horizon HTML Access client in the browser to establish the Horizon Blast session to the Unified Access Gateway instances, through the load balancer that sits in front of those instances. Starting with the October 2021 service release, for new deployments of a gateway configuration, the deployer provides the ability select either 8443 or 443 for the Blast Extreme TCP port that the deployer will configure in the corresponding Unified Access Gateway configuration. Previously, the deployer configured 443 by default without providing a choice. If your gateway configuration was deployed prior to the date of the October 2021 service release, that configuration typically has 443 port set in the Blast External URL field in its Unified Access Gateway administration settings. <strong>Note</strong> Port 8443 is preferred because it is more efficient, provides better performance, and uses less resources on the Unified Access Gateway instances. Port 443 is less efficient and less performant. Using port 443 will result in CPU congestion in the instances. Port 443 would be used in your deployment only if your organization has set client-side restrictions, such as your organization only permits 443 outbound and not 8443.</td>
</tr>
<tr>
<td>Source</td>
<td>Target</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>Horizon</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>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</td>
<td>Microsoft Azure load balancer for these Unified Access Gateway instances</td>
<td>8443</td>
<td>TCP</td>
<td>Blast Extreme via Blast Secure Gateway on Unified Access Gateway for data traffic from as Horizon Client. For a Horizon Cloud pod, this port is selected using the Blast Extreme TCP Port menu in the deployment wizard. Ensure that your networking allows the clients the outbound access to whichever one you specified for the external gateway. This URL is used by the clients to establish the Horizon Blast session to the Unified Access Gateway instances, through the load balancer that sits in front of those instances. Starting with the October 2021 service release, for new deployments of a gateway configuration, the deployer provides the ability select either 8443 or 443 for the Blast Extreme TCP port that the deployer will configure in the corresponding Unified Access Gateway configuration. Previously, the deployer configured 443 by default without providing a choice. If your gateway configuration was deployed prior to the date of the October 2021 service release, that configuration typically has 443 port set in the Blast External URL field in its Unified Access Gateway administration settings. Note Port 8443 is preferred because it is more efficient, provides better performance, and uses less resources on the Unified Access Gateway instances. Port 443, is less efficient and less performant. Using port 443 will result in CPU congestion in the instances. Port 443 would be used in your deployment only if your organization has set client-side restrictions, such as your organization only permits 443 outbound and not 8443.</td>
</tr>
<tr>
<td>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>
</tbody>
</table>
Table 6-24. 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>8443</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>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.</td>
</tr>
<tr>
<td>Browser</td>
<td>Microsoft Azure load balancer for these Unified Access Gateway instances</td>
<td>8443 or 443, depending on what was set in your deployment</td>
<td>TCP</td>
<td>Blast Extreme via Blast Secure Gateway on Unified Access Gateway for data traffic from the Horizon HTML Access client (web client). For a Horizon Cloud pod, this port is selected using the Blast Extreme TCP Port menu in the deployment wizard. Ensure that your networking allows the clients the outbound access to whichever one you specified for the external gateway. This URL is used by the Horizon HTML Access client in the browser to establish the Horizon Blast session to the Unified Access Gateway instances, through the load balancer that sits in front of those instances. Starting with the October 2021 service release, for new deployments of a gateway configuration, the deployer provides the ability select either 8443 or 443 for the Blast Extreme TCP port that the deployer will configure in the corresponding Unified Access Gateway configuration. Previously, the deployer configured 443 by default without providing a choice. If your gateway configuration was deployed prior to the date of the October 2021 service release, that configuration typically has 443 port set in the Blast External URL field in its Unified Access Gateway administration settings. <strong>Note</strong> Port 8443 is preferred because it is more efficient, provides better performance, and uses less resources on the Unified Access Gateway instances. Port 443, is less efficient and less performant. Using port 443 will result in CPU congestion in the instances. Port 443 would be used in your deployment only if your organization has set client-side restrictions, such as your organization only permits 443 outbound and not 8443.</td>
</tr>
</tbody>
</table>
Table 6-25. Internal End User Connections Ports and Protocols when using Direct 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 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, UDP</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, UDP</td>
<td>Blast Extreme</td>
</tr>
<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 Agents Installed in the Base VM, VDI Desktop VMs, and Farm RDSH VMs

The following ports must allow traffic between the agent-related software that is installed in the base VMs, desktop VMs, and farm RDSH VMs and the pod 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>Pod 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>Pod 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>
Ports and Protocols Required by App Volumes Features

As stated in App Volumes Applications for Horizon Cloud on Microsoft Azure - Overview and Prerequisites, to support the use of the App Volumes features that are supported for use with a Horizon Cloud pod, you must configure port 445 for TCP protocol traffic on the pod's tenant subnet. Port 445 is the standard SMB port for accessing an SMB file share on Microsoft Windows. The AppStacks are stored in an SMB file share located in the same resource group as the pod manager VMs.

Important If you intend to integrate your Horizon Cloud pod and NSX Cloud version 3.1.1 and later, and you want to use App Volumes assignments, you must manually open this port 445/TCP for the pod's tenant subnet in your NSX firewall rules after you deploy the NSX PCG and before you create your first App Volumes assignment using that pod.

<table>
<thead>
<tr>
<th>Source</th>
<th>Target</th>
<th>Port</th>
<th>Protocol</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>App Volumes agent in the base imported VM, the golden images, desktop VMs, farm RDSH VMs</td>
<td>SMB file share in the pod manager's resource group</td>
<td>445</td>
<td>TCP</td>
<td>On the pod's tenant subnet, for access to the App Volumes AppStacks that are stored in the SMB file share.</td>
</tr>
</tbody>
</table>

Integration with Workspace ONE Assist for Horizon - DNS, Ports, and Protocol Requirements

Workspace ONE Assist for Horizon is a product in the Workspace ONE UEM line of products. As of the August 2021 Horizon Cloud release, when specific requirements are met, you can integrate use of that product with VDI desktops provisioned from your Horizon Cloud tenant's pods. For full details of the requirements, see the Workspace ONE for Horizon and Horizon Cloud document.

Use of the assistance features requires outbound communication between the VDI desktop VMs and the Workspace ONE Assist server which supports the integration with your Horizon Cloud tenant.

DNS Requirements

Ensure that the DNS name of your Workspace ONE Assist server is resolvable and reachable from the pod's tenant subnets on which the VDI desktop VMs will reside. The Workspace ONE for Horizon and Horizon Cloud document provides the DNS names of the Workspace ONE Assist servers.

Ports and Protocol Requirements

Outbound traffic using port 443, TCP and HTTPS, must be allowed from the VDI desktop VMs on which the Workspace ONE Assist for Horizon application is installed.
When Required for An Active Support Request, Temporary Jump Box Ports and Protocols

If you make a support request to VMware and the support team determines the way to service that request is to deploy a temporary jump box VM for SSH communication with the VMware-managed appliances, that jump box requires the ports and protocols described here.

Permission will be requested from you for a support-related jump box deployment. The VMware support team will inform you of the communication requirements, as appropriate for any support situation.

This support-related jump box VM is designed to communicate as a source to the following destinations.

- The pod’s pod manager VMs’ port 22 using SSH and port 22.
- The Unified Access Gateway VMs’ port 9443 using HTTPS.
- The gateway connector VM’s port 22 using SSH, in a deployment where the external gateway is deployed in its own VNet.
- The Horizon Edge Virtual Appliance port 22 using SSH, in a deployment with Horizon Infrastructure Monitoring configured.

The nature of the support request and the appliances used in the deployment determine which specific VMware-managed appliance or appliances must be allowed as the target for the communication.

Table 6-27. Support-Related Jump Box Ports and Protocols

<table>
<thead>
<tr>
<th>Source</th>
<th>Target</th>
<th>Port</th>
<th>Protocol</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jump box VM</td>
<td>Pod manager VMs</td>
<td>22</td>
<td>SSH</td>
<td>If VMware Support requires this communication to one or more of the listed appliances to fulfill your support request, the jump box VM communicates over the management subnet to port 22 on the target appliance.</td>
</tr>
<tr>
<td></td>
<td>Gateway connector VM</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Horizon Edge Virtual Appliance</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Unified Access Gateway VMs</td>
<td>9443</td>
<td>HTTPS</td>
<td>If VMware Support requires this communication to fulfill your support request, the jump box VM communicates over the management subnet to configure settings in the Unified Access Gateway configuration.</td>
</tr>
</tbody>
</table>
Because these VMs are assigned IP addresses dynamically, the following network rules can provide for the described communications. During the support-request activities, always seek guidance and supervision from VMware Support for requirements for the support-related jump box deployment.

- 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.

**True SSO and Certificate Management and Horizon Cloud on Microsoft Azure deployments**

The Horizon Cloud pod-provisioned desktop VMs do not communicate directly with the Enrollment Server. The Horizon Cloud on Microsoft Azure deployment’s active pod manager VM relays the certificate requests to the Enrollment Server. Once the certificate is obtained, the Horizon agent in the desktop VMs uses that certificate to perform the Certificate Logon operation on behalf of the desktop user.

The request-response architecture is the same for a Horizon Cloud on Microsoft Azure deployment’s pod manager VM as it is for a Horizon deployment’s Horizon Connection Server. In a Horizon Cloud on Microsoft Azure deployment, the pod manager VMs have connections to the desktop VMs on the primary VM subnet (also called the tenant subnet), and on any additional VM subnets that the VDI administrator might have added using the Edit Pod workflow.

Two classes of certificate will be validated by various components: user certificates and channel certificates. True SSO adds a user certificate that is validated by the authentication server. In this case of a Horizon Cloud on Microsoft Azure deployment, that authentication server is a Microsoft Active Directory server. Because the Microsoft architecture determines which port numbers can be used for this certificate validation, a wide range of port numbers can be used for this validation since the ports are part of the Microsoft architecture itself and not specific to the Horizon Cloud on Microsoft Azure deployment itself.

When using True SSO in a Horizon Cloud on Microsoft Azure deployment, the Horizon agent generates a CSR and sends it to the deployment’s active pod manager VM over the communication channel already in place between that pod manager VM and that Horizon agent. The pod manager VM relays the request to the Enrollment Server over a secure SSL-encrypted TCP channel (port 32111 or the port configured by the customer in the Enrollment Server installation). The Enrollment Server generates a CMC request, appends the CSR and the user name as provided by the Pod Manager, signs the CMC using the Enrollment Agent Certificate, and submits it to the Certificate Authority using the MS-DCOM (RPC) protocol.

The Horizon agent receives the certificate, serializes it as a logon credential, and submits it to the Windows logon process. The LSASS Windows component receives the certificate, validates it (checks that it is valid and trusted, and that the local machine holds the private key for the certificate) and then sends it to a Domain Controller (DC). The DC may choose to check the CRL as specified in the user certificate.
Visually Rich Networking Diagrams
For a visually rich depiction of the relationships between these components, ports, and protocols, see the VMware Digital Workspace Tech Zone's network diagrams and descriptions at https://techzone.vmware.com/resource/vmware-horizon-cloud-service-microsoft-azure-network-ports-diagrams.

Create a Horizon Cloud App Registration in the Pod's Subscription
For Horizon Cloud Service on Microsoft Azure deployments, the service uses API calls to deploy the pod into a Microsoft Azure subscription and manage that pod and the pod-provisioned VDI desktops and farms. To provide the ability for Horizon Cloud to use its API calls in the pod's subscription, you create an app registration.

Brief Introduction
For the initial deployment of the pod, the pod deployer calls the APIs in the Microsoft Azure subscription that you have chosen to use for the pod. These API calls perform actions in the pod's subscription to create items such as the pod manager VM, the VM's NICs, the network security groups (NSGs) on those NICS — all of the resources that a Horizon Cloud pod requires.

Then, after pod deployment, Horizon Cloud must continue to have the ability call APIs in the pod's subscription. Post-pod-deployment, the service uses API calls to create the base image VMs for the golden images, run sysprep on the golden images, create farm hosts and VDI desktop VMs, add and edit the pod's gateway configurations, and to maintain and upgrade the pod.

Create the App Registration Before Running the Pod Deployer
Because the pod deployer needs to call the APIs during the pod deployment process for programmatically creating the pod's resources within the pod's subscription, the app registration and client secret key must exist before you start the deployment wizard. Creation of the app registration automatically creates a service principal object in the pod subscription.

The client secret key must be generated in the Azure Portal and a role assigned to the Horizon Cloud app registration to operate at the level of the pod's subscription.

If you want to use the feature where the external Unified Access Gateway configuration is deployed in its own subscription, separate from the pod's subscription, Horizon Cloud must also have the ability to call APIs in that subscription at the time you run the wizard to deploy that external gateway. In this case, an app registration and client secret key are needed in that subscription in addition to the ones for the pod's subscription.

About Assigning a Role to the App Registration
The Horizon Cloud app registration must have an assigned role in the pod's subscription. Typically the built-in Contributor role is the role used by Horizon Cloud with the pod's subscription. The reason why the Contributor role is used is because this role covers all of the API calls that Horizon Cloud would need to perform within the pod's subscription.
The role assignment must be a direct assignment. The use of a group-based assignment of a role — in which the role is assigned to a group and the app registration is a member in that group — is currently unsupported.

If your organization prefers to avoid use of the Contributor role in the pod’s subscription, Horizon Cloud also supports use of a custom role instead. If used, the custom role needs to provide for the specific API calls that Horizon Cloud needs to use. For more information, see the Custom Roles and the Horizon Cloud App Registration section near the bottom of this page.

Register Resource Providers

In the pod’s subscription, the following resource providers must all have Registered status. You might see that some of the resource providers in this list already have Registered status while others do not. That is a result of standard Microsoft Azure behavior, where they have a set of resource providers typically registered for all Azure subscriptions.

You will want to ensure these listed resource providers have Registered status before running the pod deployment wizard. At the wizard’s final step, it will validate that these resource providers have Registered status and will prevent the start of the pod deployment if one is unregistered.

- Microsoft.Compute
- microsoft.insights
- Microsoft.Network
- Microsoft.Storage
- Microsoft.KeyVault
- Microsoft.Authorization
- Microsoft.Resources
- Microsoft.ResourceHealth
- Microsoft.ResourceGraph
- Microsoft.Security
- Microsoft.DBforPostgreSQL
- Microsoft.Sql
- Microsoft.MarketplaceOrdering

The following screenshot is an illustration of seeing the Registered status and unregistered status in the Azure portal.
To verify the resource providers in the pod’s subscription:

1. Log in to the Azure portal and search for the subscription into which you plan to deploy the pod.

2. Click the subscription name and scroll down until you see (Resource providers).

3. Look for the resource providers in the preceding list and verify that they each display (Registered) status.

   For any resource provider from the preceding list that you see as NotRegistered, use the portal to register it.

Creating the Horizon Cloud App Registration

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

When you are going to use the Horizon Cloud feature where the external gateway uses its own subscription, separate from the pod’s, you would repeat the steps in that subscription for its app registration.
To complete all of the following steps in the Azure Portal yourself, your portal login must have sufficient permissions to create an app registration and assign a role to that app registration in the subscription into which you plan to deploy the pod. If you are not the owner or administrator of that subscription, ask one of them if you have the required permissions to create an app registration and assign a role to that app registration.

1. Log in to the Microsoft Azure portal using credentials that have the ability to register applications.

2. In the portal’s search bar, search for App registrations, and click App registrations when you see it appear in the results list.

The portal displays the App registrations page.

3. On the App registrations page, click New registration.

4. Type a display name that will remind you this registration is for Horizon Cloud use.

5. Select Accounts in this organizational directory only.

6. Leave the optional Redirect URI section in its default, empty state.

7. Click the Register button to complete creating the app registration.

The newly created app registration is displayed on screen.

8. Copy the application ID and directory ID and save them to a location where you can retrieve them later when you run the deployment wizard. The following screenshot illustrates an app registration named Hzn-Cloud-Principal and a green arrow pointing to where the application ID and directory ID are displayed.
9 Then create the app registration’s client secret key:

a In the preceding screenshot, see where **Certificates & secrets** appears. In the Azure portal, on your newly created app registration page, click **Certificates and secrets**.

b Click **New client secret**.

c As illustrated in the following screenshot, the portal displays the **Add a client secret** screen. Type a description, select an expiration duration, and click **Add**. The key description must be 16 characters or less, for example `Hzn-Cloud-Key1`.

![Add a client secret](image)

**Important** Keep this screen open until you copy the secret value and paste the value into a location where you can retrieve it later.

<table>
<thead>
<tr>
<th>DESCRIPTION</th>
<th>EXPIRES</th>
<th>VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hzn-cloud-key1</td>
<td>7/1/2022</td>
<td><img src="image" alt="Secret Value" /></td>
</tr>
</tbody>
</table>

d Copy the secret value to a location where you can retrieve it later when you run the deployment wizard. The wizard has a field into which you paste this value.

10 Add a role assignment to the Horizon Cloud app registration. Assign the role at the subscription level:

a Navigate to the 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 into which you plan to have the pod deployer deploy the pod.

**Note** At this point, from the screen, note the subscription ID which you will later need in the deployment wizard.
b Click Access control (IAM) and then click Add role assignment to open the Add role assignment screen.

c In the Add role assignment screen, for Role, select the Contributor role.

If your organization has said they prefer that a custom role be used for Horizon Cloud, select the custom role that your organization set up for this purpose.

d In Assign access to drop-down list, select Azure AD user, group, or application.

e Use the Select box to search for the Horizon Cloud app registration's name. The following screenshot illustrates this step.

```
Role

Contributor

Assign access to

Azure AD user, group, or application

Select

Hzni

Hzn-Cloud-Principal
```

f Click the name that you gave to your created Horizon Cloud app registration to make it a selected member and then click Save.
Summary

At this point, you've created and configured the Horizon Cloud app registration, confirmed the registration status of the resource providers that Horizon Cloud requires, and you have the subscription-related values you will need to enter into 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

**Note**  Horizon Cloud cannot detect or know what expiration duration you set for the app registration's client secret key. To ensure that Horizon Cloud can continue using this app registration to make its necessary API calls to manage the pod and its resources, you must remember to refresh the key before the key's expiration date is reached and then enter the new key into your Horizon Cloud environment. Currently, the longest expiration duration that one can set using the Microsoft Azure portal is two (2) years. If the key expires at the end of two years and you have not refreshed it or entered new key information into your Horizon Cloud environment for use with the pod, the pod associated with the expired key will stop working. If you prefer to create a secret key with a lifetime longer than the two years that the Microsoft Azure portal's user interface allows, Microsoft Azure currently provides that ability using PowerShell, Azure CLI, or Graph API.

**Custom Roles and the Horizon Cloud App Registration**

When your organization prefers to avoid use of the **Contributor** role in the pod's subscription, your organization can create a custom role instead and have that assigned to the Horizon Cloud app registration. The custom role must be configured so that it permits the API calls required by Horizon Cloud. If your organization prefers to avoid use of the **Contributor** role in the pod's subscription, refer to the information in When Your Organization Prefers to Use a Custom Role for the Horizon Cloud App Registration.
When Your Organization Prefers to Use a Custom Role for the Horizon Cloud App Registration

So that Horizon Cloud app registration has the ability to make API calls in the pod's subscription — or in the optional external gateway subscription — and perform its VDI-related operations, a role must be assigned to it. Usually the Contributor role is used for this purpose. For organizations that prefer to avoid use of the Contributor role, they can create a custom role, and have the custom role fulfill the purpose of giving the Horizon Cloud app registration the ability to perform the required API calls.

In addition to a custom role for the Horizon Cloud app registration in the pod's subscription, if your organization prefers to adopt the approach to have a separate subscription for the pod's external Unified Access Gateway configuration and also select to have the gateway resources deployed into a specific resource group that your organization sets up for that purpose, the custom role for that gateway's subscription can have more granular, narrow-scope permissions than the custom role for the pod's subscription.

Brief Introduction to Custom Roles

The overarching concept is that Horizon Cloud needs to perform certain operations in the pod's 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.

In Microsoft Azure, a role provides for a set of management operations that can be performed by an app registration's service principal. A management operation is a combination of the resource and action performed on that resource.

You can restrict the Horizon Cloud app registration's abilities in the pod's subscription and (optional) gateway's subscription to the minimum operations required, by following the rules described below.

Overview of the Available Use Cases

When discussing Horizon Cloud required operations in subscriptions and resource groups, there are these use cases.

Note In the two-subscription use case, the role for the app registration in the pod's subscription 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 pod’s 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 pod’s subscription.</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>
<tr>
<td>■ One subscription specified to use for the external Unified Access Gateway configuration’s resources</td>
<td></td>
</tr>
<tr>
<td>■ One subscription for the rest of the pod resources</td>
<td></td>
</tr>
<tr>
<td>Use Case</td>
<td>Description</td>
</tr>
<tr>
<td>----------</td>
<td>-------------</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 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 your VNet has custom routes. Microsoft Azure Cloud has a feature called custom routes. | If your VNet has custom routes, a permission is needed in addition to all of the ones for the above use cases: `Microsoft.Network/routeTables/join/action`. |
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. The custom role must permit the operations in the following table. The * (wild card character) grants access to all operations that match the string within the listed operation.

Table 6-28. 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>Microsoft.Compute/*/read</td>
<td><a href="https://docs.microsoft.com/en-us/azure/role-based-access-control/resource-provider-operations#microsoftcompute">https://docs.microsoft.com/en-us/azure/role-based-access-control/resource-provider-operations#microsoftcompute</a></td>
</tr>
</tbody>
</table>
Table 6-28. 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 6-28. 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.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>
</tbody>
</table>
Table 6-28. 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 6-28. 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>Operation</td>
<td>Description in the Microsoft Azure Documentation</td>
</tr>
<tr>
<td>-----------</td>
<td>--------------------------------------------------</td>
</tr>
</tbody>
</table>
Table 6-28. 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.Compute/galleries/write</td>
<td></td>
</tr>
<tr>
<td>Microsoft.Compute/galleries/delete</td>
<td></td>
</tr>
<tr>
<td>Microsoft.Compute/galleries/images/*</td>
<td></td>
</tr>
<tr>
<td>Microsoft.Compute/galleries/images/versions/*</td>
<td></td>
</tr>
<tr>
<td>Microsoft.MarketplaceOrdering/offertypes/publishers/offers/plans/agreements/write</td>
<td></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. The ID is the unique ID of the custom role. When Azure PowerShell or Azure CLI is used to create a custom role, this ID is automatically generated. For the variable `mysubscriptionId1`, substitute the ID of the subscription in which the custom role will be used — the pod’s subscription or the optional gateway subscription.
Table 6-29. Sample JSON for a Role Permitting the Horizon Cloud Required Operations When Assigning Permissions at the Subscription Level

```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.Resources/deployments/**",
    "Microsoft.Storage/*/read",
    "Microsoft.Storage/storageAccounts/**",
    "Microsoft.Compute/galleries/read",
    "Microsoft.Compute/galleries/write",
    "Microsoft.Compute/galleries/delete",
    "Microsoft.Compute/galleries/images/**",
    "Microsoft.Compute/galleries/images/versions/**",
    "Microsoft.MarketplaceOrdering/offertypes/publishers/offers/plans/agreements/read",
    "Microsoft.MarketplaceOrdering/offertypes/publishers/offers/plans/agreements/write"
  ],
  "NotActions": [],
  "DataActions": [],
  "NotDataActions": [],
  "AssignableScopes": [
    "subscriptions/mysubscriptionId"
  ]
}
```

When Custom Routes are in Your VNet and Its Subnets

Microsoft Azure cloud has a feature called custom routes.

If you have such routes added to your VNet and its subnets, this additional permission is needed.
<table>
<thead>
<tr>
<th>Operation</th>
<th>Description in the Microsoft Azure Documentation</th>
</tr>
</thead>
</table>

### 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, at the subscription level of the external gateway's subscription, your organization can use the built-in Reader role for Horizon Cloud app registration to use, plus a custom role at the level of the named resource group.

Your organization would create a custom role which specifies the permissions in the following table. That custom role would then be assigned to Horizon Cloud app registration to operate with the specifically named resource group in the external gateway's subscription. You or your organization would pre-create that named resource group in the subscription in which you are deploying the external gateway.

Some specific permissions on subnets and on the VNet are also required, depending on your planned deployment options:

- If this external gateway deployment will use subnets that are created 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.

Table 6-31. 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>Microsoft.Compute/*/read</td>
<td><a href="https://docs.microsoft.com/en-us/azure/role-based-access-control/resource-provider-operations#microsoftcompute">https://docs.microsoft.com/en-us/azure/role-based-access-control/resource-provider-operations#microsoftcompute</a></td>
</tr>
</tbody>
</table>
Table 6-31. 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>
</table>
Table 6-31. 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.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>Microsoft.KeyVault/*/read</td>
<td><a href="https://docs.microsoft.com/en-us/azure/role-based-access-control/resource-provider-operations#microsoftkeyvault">https://docs.microsoft.com/en-us/azure/role-based-access-control/resource-provider-operations#microsoftkeyvault</a></td>
</tr>
</tbody>
</table>
Table 6-31. Microsoft Azure Resource Operations that Must Be Permitted on the Specified Resource Group (continued)

<table>
<thead>
<tr>
<th>Operation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<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>
</tbody>
</table>
Table 6-31. 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>
</table>
Table 6-31. 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.MarketplaceOrdering/offertypes/publishers/offers/plans/agreements/write</td>
<td></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 a Horizon Cloud App Registration in the Pod’s Subscription. 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.
<table>
<thead>
<tr>
<th>Required Value</th>
<th>How to Collect</th>
<th>Your Values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Environment</td>
<td>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.</td>
<td></td>
</tr>
<tr>
<td>Subscription ID</td>
<td>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.</td>
<td></td>
</tr>
<tr>
<td>Directory ID</td>
<td>In the Microsoft Azure portal, click Azure Active Directory &gt; Properties (under Manage).</td>
<td></td>
</tr>
<tr>
<td>Application ID</td>
<td>In the Microsoft Azure portal, click Azure Active Directory &gt; App registrations, and then click the application registration that you created for Horizon Cloud using the steps in Create a Horizon Cloud App Registration in the Pod’s Subscription.</td>
<td></td>
</tr>
<tr>
<td>Application Key</td>
<td>Obtain the key when you generate it in the Microsoft Azure portal. See Create a Horizon Cloud App Registration in the Pod’s Subscription.</td>
<td></td>
</tr>
</tbody>
</table>

**Resource Providers That Horizon Cloud Requires to Be in Registered Status in the Microsoft Azure Subscription**

To provide for creating and managing a Horizon Cloud pod in your Microsoft Azure subscription, Horizon Cloud requires the ability to create and manage various resources in that subscription. Therefore, your subscription must have specific resource providers in Registered status so that Horizon Cloud can perform its required operations to initially deploy the pod and to manage and upgrade the pod over its lifetime. This documentation article lists the resource providers that Horizon Cloud needs in Registered status.

According to the Microsoft Azure documentation, a resource provider is a service that supplies the Microsoft Azure resources, such as the Microsoft.KeyVault resource provider supplies the key vault type of resource. As described in the Microsoft Azure documentation topic Azure resource providers and types, before a particular resource can be used in a subscription, that subscription must be registered for the resource provider that supplies that type of resource. That Microsoft Azure documentation goes on to state that some resource providers are registered by default while others require explicit manual registration in the subscription.
Most importantly before deploying a pod, verify that the resource providers listed here all show **Registered** status in the subscription. When deploying a new pod, the pod deployer creates resources in your subscription by issuing commands to the various Microsoft Azure resource providers to request creation of the types of resources that the deployer needs to create. If a resource provider that requires your explicit manual registration is not in **Registered** status before the pod deployment reaches its **Validate & Proceed** step, the wizard will block the pod deployment from going forward from that point. In the **Validate and Proceed, and then Start the Pod Deployment Process**, the pod deployer will validate whether its required set of resource providers has the **Registered** status in the subscription. If a resource provider that the pod deployer requires does not have the **Registered** status, the wizard displays an error message. The following screenshot is an example of the case where the Microsoft.Security resource provider is not in **Registered** status in the subscription.

![Pod configuration validation error](image)

**Horizon Cloud Required Resource Providers**

- Microsoft.Compute
- microsoft.insights
- Microsoft.Network
- Microsoft.Storage
- Microsoft.KeyVault
- Microsoft.Authorization
- Microsoft.Resources
- Microsoft.ResourceHealth
- Microsoft.ResourceGraph
- Microsoft.Security
- Microsoft.DBforPostgreSQL
- Microsoft.Sql
- Microsoft.MarketplaceOrdering
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-----
-----BEGIN CERTIFICATE-----
.... (the intermediate CA certificate)
-----END CERTIFICATE-----
-----BEGIN CERTIFICATE-----
.... (the trusted root certificate)
-----END CERTIFICATE-----
-----BEGIN RSA PRIVATE KEY-----
.... (your server key from mycaservercertkeyrsa.pem)
-----END RSA PRIVATE KEY-----
```

**Note**  The server certificate should come first, followed by any intermediate ones, and then the trusted root certificate.

5 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.

**Using the Horizon Universal Console to Perform an Automated Deployment of a Pod into Microsoft Azure**

You run the pod deployment wizard to deploy the components that together make up a pod-manager-based pod on Microsoft Azure 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.

**Important**  This wizard deploys the pod-manager-based type of pod into Microsoft Azure. The wizard does not deploy a Horizon pod on Azure VMware Solution (AVS), which uses the Horizon Connection Server technology. For information about manually deploying a Horizon pod on AVS, see Tech Zone's [Horizon on Azure VMware Solution Architecture](#) and [Deploying Horizon with Azure VMware Solution](#). For that type of pod, after you have manually deployed such a pod, you can connect it to Horizon Cloud using the Horizon Cloud Connector and the steps in [High-Level Workflow When You are Onboarding an Existing Manually Deployed Horizon Pod](#).
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
- Deploying with an external or internal Unified Access Gateway configuration, deploying with both, or deploy with neither and add them later. 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>
<tr>
<td>None</td>
<td>Either type or both types</td>
</tr>
</tbody>
</table>

- Deploying with an external Unified Access Gateway configuration in its own VNet, separate from the pod’s VNet.

This scenario has these additional variations:

- 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.

- When deploying with an external Unified Access Gateway configuration in its own subscription, you can also choose to deploy it in a named resource group in that separate subscription. In this case, you must pre-create that resource group in the subscription before running the Add Pod wizard.

- Deploying with the option for two-factor authentication configured on the pod’s gateway configurations. If you deploy without two-factor authentication 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 for Deploying a Pod-Manager-Based Pod**
   This wizard is used to automatically deploy a pod-manager-based pod into your Microsoft Azure subscription. When using this wizard to deploy your very first pod, you start the pod deployment wizard using the **Manage > Add Pod** feature in the **Microsoft Azure** row of the Horizon Universal 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 Using the Deployment Wizard**
   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-manager-based pod with one or more gateways configured. Unified Access Gateway provides the gateway environment for this type of pod.
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:

- 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 create the Azure StorageV2 account type in the pod’s resource group in the subscription. This storage account is used for the pod’s App Volumes features. During the pod deployment process, ensure that your Microsoft Azure Policies do not restrict or deny the creation of content requiring the Azure StorageV2 account type.

- 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 on 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, NVv4-series, and NCv2-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 and Related Service Features and Horizon Cloud Pod - Ports and Protocols Requirements.

- 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.

  **Important** Editing or updating of the proxy settings on a pod after the pod is deployed in Microsoft Azure is currently unsupported. Also, adding a proxy configuration to a deployed pod that was deployed without proxy settings is currently unsupported.

- 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 reuse existing subnets that already have items that are using IP addresses on those subnets. If an IP address is already in use on the subnets, issues such as the pod failing to deploy and other downstream IP-conflict issues have a high likelihood of occurring. Do not put any resources on these subnets or otherwise use any of the IP addresses. This caution notice includes pods deployed from Horizon Cloud — do not reuse subnets on which you already have a deployed pod.
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. If you are planning to use the same FQDN for both the external and internal gateway configurations, after the pod is deployed, you must configure the incoming end-user client traffic to route to the appropriate gateway 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. In this scenario where 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 prevent 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

**Note** Deploying an external gateway using its own VNet will deploy a gateway connector VM. In Ports and Protocols Requirements for a Horizon Cloud Pod, the section that describes the gateway connector VM's ports and protocols also has a description of this gateway connector VM, which states this gateway connector VM has a name that contains a part like `vmw-hcs-ID`, where `ID` is the gateway's deployer ID, and a `node` part.

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 When Your Organization Prefers to Use a Custom Role for the Horizon Cloud App Registration.
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 from your authentication server’s configuration, so that you can provide it in the Add Pod wizard’s required fields.

Obtain the following listed information, according to the type you have.

**RADIUS**

If you are configuring settings for both a primary and auxiliary RADIUS 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 1812/UDP for RADIUS.
- 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.

**RSA SecurID**

**Note** The RSA SecurID type is supported with Horizon Cloud on Microsoft Azure deployments that are running manifest 3139.x or later. The UI option to specify the RSA SecurID type in the Add Pod and Edit Pod wizards will become visible to select in the wizards starting in mid-March 2022.

- Access key from the RSA SecurID authentication manager server.
- RSA SecurID communication port number. Typically 5555, as set in the RSA Authentication Manager system settings for RSA SecurID Authentication API.
- Host name of the RSA SecurID authentication manager server.
- IP address of that RSA SecurID authentication manager server.
If the RSA SecurID authentication manager server or its load balancer server has a self-signed certificate, you will need the CA certificate to provide in the Add Pod wizard. The certificate should be in PEM format (file types .cer or .cert or .pem)

Start the Pod Deployment Wizard for Deploying a Pod-Manager-Based Pod

This wizard is used to automatically deploy a pod-manager-based pod into your Microsoft Azure subscription. When using this wizard to deploy your very first pod, you start the pod deployment wizard using the Manage > Add Pod feature in the Microsoft Azure row of the Horizon Universal Console's Getting Started page.

**Note**  Login authentication into the cloud-based console relies on authenticating account credentials with VMware Cloud Services. If that service is unable to complete the necessary authentication requests, then 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.workspaceone.com](https://status.workspaceone.com) to see the latest system status. On that page, you can also subscribe to receive updates.

**Important**  This wizard does not support deploying a Horizon pod into Azure VMware Solution (AVS). In the current release, there is no wizard that automates deployment of a pod into Azure VMware Solution. For that type of pod, you must first manually stand up the Horizon pod in AVS and then use the Horizon Cloud Connector to connect it to Horizon Cloud. For information about standing up a Horizon pod in AVS, see Tech Zone's Horizon on Azure VMware Solution Architecture.

**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 using one of the following methods:

   - Unless otherwise directed by your organization, you can go to the Horizon Cloud Service address at [https://cloud.horizon.vmware.com](https://cloud.horizon.vmware.com). The system then automatically redirects to the standard VMware Cloud Services login screen at [https://console.cloud.vmware.com](https://console.cloud.vmware.com). Sign in using your VMware Customer Connect account's credentials. (The previous name for the VMware Customer Connect account was My VMware account). The account credentials are the primary email address, such as user@example.com, and the password that are set in the account's profile.

   - If your organization has asked you to access their Horizon Cloud tenant using VMware Cloud Services, first log in to VMware Cloud Services at [https://console.cloud.vmware.com](https://console.cloud.vmware.com). When logged in to VMware Cloud Services, then click the Horizon Cloud card that appears under the My Services section.
If organization has asked you to access their Horizon Cloud tenant using their Workspace ONE environment, you can first log in to that Workspace ONE console, and then use the Horizon Cloud card that appears in your set of services.

The following screenshot illustrates the VMware Cloud Services login screen.

If you have not previously accepted the Horizon Cloud terms of service using those 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.
**Note**  As described in Horizon Service Deployments and Onboarding Pods — Horizon Pods and Horizon Cloud Pods on Microsoft Azure, depending on the configuration of the Horizon Cloud tenant record, you might see a blue banner about onboarding to VMware Cloud Services. That step is not required for pod deployment and can be performed later. For information about that feature, see Onboard Your Horizon Cloud Tenant to VMware Cloud Services.

2. In the Add Cloud Capacity row, click **Manage > Add Pod**.

The Add Cloud Capacity wizard opens to its first step.
3 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 Using the Horizon Universal Console to Perform an Automated Deployment of a 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 for Deploying a Pod-Manager-Based Pod.
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.
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><strong>Apply</strong></td>
<td>Select the name of a previously entered subscription or select <strong>Add New</strong> to enter new subscription information.</td>
</tr>
<tr>
<td><strong>Subscription</strong></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>
</tbody>
</table>
| **Environment** | 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  
<p>| <strong>Subscription ID</strong> | 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. |
| <strong>Directory ID</strong> | 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. |
| <strong>Application ID</strong> | 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. |</p>
<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Application Key</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>Use a Different Subscription for External Gateway</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** This screen does not provide a way to delete the 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 within this specific screen to change the values, or edit or delete that subscription name to start over with the same name. If this happens to you, first cancel out of the wizard. Then navigate to the console’s Getting Started page, click **Manage > Manage Subscriptions**, and use the **Delete** action to delete that previously entered subscription name. Then you can start the new-pod wizard again, enter the 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**

- **Apply Subscription:** Add New
- **Subscription Name:** mysub1
- **Environment:** Azure - Commercial
- **Subscription ID:**
- **Directory ID:**
- **Application ID:**
- **Application Key:**

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 When Your Organization Prefers to Use a Custom Role for the Horizon Cloud App Registration.

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 Using the Deployment Wizard**.

**Specify Pod Configuration Information for the Horizon Cloud Pod You Are Deploying Into Microsoft Azure Using the Deployment Wizard**

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.

**Remember** This wizard deploys the pod-manager-based type of pod. The console currently does not provide a deployment wizard to deploy a Horizon pod in Azure VMware Solution (AVS). For information about deploying an Horizon pod in AVS, see Tech Zone’s [Horizon on Azure VMware Solution Architecture](#).

**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 that 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.
<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pod Name</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>Location</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. Currently, due to a known issue, the location names are not localized.</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. <strong>Important</strong> 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, NVv4-series, and NCv2-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>Azure Resource Tags</td>
<td>Optional: Create custom tags to be applied to Azure resource groups. Azure resource tags are only applied to the resource groups, and are not inherited by the resources in the groups. To create the first tag, enter information in the Name and Value fields. To create an additional tag, click + and then enter information in the Name and Value fields that appear below the existing ones. - You can create a maximum of 10 tags. - The tag name is limited to 512 characters, and the tag value is limited to 256 characters. For storage accounts, the tag name is limited to 128 characters, and the tag value is limited to 256 characters. - Tag names cannot contain the following characters: `&lt; &gt; % &amp; \ ? / - Tag names cannot contain these case-insensitive strings: ‘azure’, ‘windows’, ‘microsoft’ - Tag names and tag values can only contain ASCII characters. Blank spaces and characters outside the standard 128-character ASCII set (also known as high ASCII or extended ASCII characters) are not allowed.</td>
</tr>
<tr>
<td>Virtual Network</td>
<td>Select a virtual network from the list. Only virtual networks (V Nets) that exist in the region selected in the Microsoft Azure Region field are shown here. You must have already created the V Net you want to use in that region in your Microsoft Azure subscription.</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Use Existing Subnet</strong></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 <strong>Yes</strong>, the wizard’s fields for specifying subnets change to drop-down selection menus.</td>
</tr>
<tr>
<td><strong>Important</strong></td>
<td>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 <strong>Yes</strong>, you must select from existing subnets for all the pod’s required subnets.</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>Important</strong></td>
<td>* Select a subnet that has the <strong>Microsoft.SQL</strong> 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><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>Important</strong></td>
<td>* Select a subnet that has the <strong>Microsoft.SQL</strong> 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><strong>Caution</strong></td>
<td>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>Option</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>-------------</td>
</tr>
<tr>
<td>NTP Servers</td>
<td>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.</td>
</tr>
</tbody>
</table>
| 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:  
Proxy:       proxy.example.com  
Port:        3128  
Username:     
Password:     
Verify Password:  
```

The following screenshot is an example with this step completed when using subnets that are created on the VNet in advance. In this example, a proxy was not needed to meet the outbound Internet connectivity requirement. The network-related names have been redacted.
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-manager-based pod with one or more gateways configured. Unified Access Gateway provides the gateway environment for this type of pod.

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 switch off this public IP toggle in the wizard, the wizard displays a field in which you must specify an IP address. In that type of configuration, the PCoIP connections to the gateway from the Horizon clients will use this IP address.

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.
Decide what VM 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. If you anticipate your environment to scale to 2,000 sessions per pod, select F8s_v2. As stated in VMware Horizon Cloud Service on Microsoft Azure Service Limits, the A4_v2 VM model is only sufficient for proofs-of-concept (PoCs), pilots, or smaller environments where you know that you will not exceed 1,000 active sessions on the pod.

**Important** Choose the VM model wisely. In the current service release, the VM model used by the deployed instances cannot be easily changed after the gateway configuration is deployed. Changing the VM model after deployment involves deleting the gateway configuration and re-deploying it.

**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.

When you will select an external gateway configuration, Horizon Cloud expects that the FQDN specified for the external gateway configuration is publicly resolvable. If you switch off the Enable Public IP? toggle in the wizard to specify an IP address from your firewall or NAT setup, then you are responsible for ensuring this FQDN is assigned to that IP address in your firewall or NAT setup. This FQDN is used for PCoIP connections to the gateway.

If your tenant is configured with Universal Broker that has two-factor authentication configured, you must configure an external Unified Access Gateway with two-factor authentication settings.
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. Note Leaving the default enabled setting is recommended. When this toggle is switched off, clients must either connect through Workspace ONE Access with its connector appliance integrated directly to the pod managers, or the clients connect directly to the pod managers’ load balancer, or they connect through an internal gateway configuration. In the first two scenarios mentioned, of clients connecting through Workspace ONE Access integrated with the pod or connecting directly to the load balancer, some post-deployment steps will be required. In those scenarios, after the pod is deployed, upload SSL certificates to the pod manager VMs by following the steps in Configure SSL Certificates Directly on the Pod Manager VMs.</td>
</tr>
<tr>
<td>FQDN</td>
<td>Enter the required fully qualified domain name (FQDN), such as ourOrg.example.com, which the pod deployer will specify in the configuration for the gateway’s Unified Access Gateway instances. You must own that domain name and have a certificate in PEM format that can validate that FQDN. Horizon Cloud expects that this FQDN specified for the external gateway configuration is publicly resolvable. If you switch off the Enable Public IP? toggle to specify an IP address from your firewall or NAT setup, then you are responsible for ensuring this FQDN is assigned to that IP address in your firewall or NAT setup. This FQDN is used for PCoIP connections to the gateway. Important 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 a two-factor authentication server that is located outside of the VNet topology into which the Unified Access Gateway instances are deployed, you would specify the address of a DNS server that can resolve the host name of that authentication server. As an example, when your two-factor authentication is located on-premises, enter a DNS server that can resolve the name of that authentication server. As described in the Prerequisites for All Deployments, the VNet topology used for the Horizon Cloud on Microsoft Azure deployment must be able to communicate with your DNS server that you want providing external name resolution during the deployment of the Unified Access Gateway instances and for their ongoing operations. By default, the DNS server that is configured on the VNet into which the instances are deployed is the one used. When you specify addresses in DNS Addresses, the deployed Unified Access Gateway instances use those addresses in addition to the DNS server information in your VNet’s configuration.</td>
</tr>
</tbody>
</table>
### Option  Description

**Routes**  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 for communication with two-factor authentication servers.

When configuring this pod to use two-factor authentication with an on-premises authentication server, you must enter the correct route the Unified Access Gateway instances can use to reach that server. For example, if your on-premises authentication 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 Horizon Cloud on Microsoft Azure deployment.

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.

**VM Model**  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.

**Important** In the current service release, the VM model used by these instances cannot be easily changed after the gateway configuration is deployed in your subscription. Changing the VM model after deployment requires deleting the gateway configuration and re-deploying it. If you anticipate your environment to scale to 2,000 sessions per pod, select **F8s_v2**. As stated in VMware Horizon Cloud Service on Microsoft Azure Service Limits, the **A4_v2** VM model is only sufficient for proofs-of-concept (PoCs), pilots, or smaller environments where you know that you will not exceed 1,000 active sessions on the pod.

**Certificate**  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.

**Blast Extreme TCP Port**  Select the TCP port to use in the Blast Extreme TCP setting within the Unified Access Gateway configuration. That setting relates to the Blast Extreme via Blast Secure Gateway on Unified Access Gateway for data traffic sent by the client. Port 8443 is preferred because it is more efficient, provides better performance, and uses less resources on the Unified Access Gateway instances. For those reasons, the wizard has 8443 as the default value. The other choice, 443, is less efficient, less performant, and causes CPU congestion in the instances which can cause observable traffic delays in the end-user clients. The 443 choice should be used only if your organization has set client-side restrictions, such as your organization only permits 443 outbound.

**Note** The UDP port used for Blast Extreme is unaffected by this setting, and is always UDP 8443.

**Cipher Suites**  Although in almost all cases the default settings do not need to be changed, Unified Access Gateway provides this feature for optionally specifying the cryptographic algorithms used to encrypt communications between clients and the Unified Access Gateway appliances. At least one cipher suite must be selected from the on-screen list. The on-screen list displays the cipher suites allowed for the Horizon Cloud on Microsoft Azure deployment.

Specify the settings for this gateway’s Microsoft Load Balancer.
<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enable Public IP?</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>Important</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 switch off this toggle, the field <strong>Public IP for Horizon FQDN</strong> appears.</td>
</tr>
<tr>
<td>Public IP for Horizon FQDN</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 to which you are assigning the FQDN that you specified in the <strong>FQDN</strong> field. Your end users’ Horizon clients will use this FQDN 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.
### Option Description

<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>
</tbody>
</table>

**Use a Different Virtual Network — Switched off**

When the toggle is switched off, the external gateway will be deployed into the pod's VNet. In this case, you must specify the DMZ subnet.

- **DMZ Subnet** - 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.

  **Important** 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.

- **DMZ Subnet (CIDR)** - When **Use Existing Subnet** is switched off 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.

**Use a Different Virtual Network — Enabled**

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.

  **Important** 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.

In this case, the gateway's VNet and pod's VNet are peered. The best practice is to have the subnets created in advance, and not use the CIDR entries here. See **Prerequisites When Deploying With an External Unified Access Gateway Configuration Using its Own VNet or Subscription Separate from the Pod's VNet or Subscription**.

- **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.

- **Back-end subnet** - Specify the subnet to use for the gateway's back end subnet. A CIDR of at least /27 is required.

- **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.

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 <code>ourOrg.example.com</code>, 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. <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 Unified Access Gateway configuration to use two-factor authentication with a two-factor authentication server that is located outside of the VNet topology into which the Unified Access Gateway instances are deployed, you would specify the address of a DNS server that can resolve the host name of that authentication server. As an example, when your two-factor authentication is located on-premises, enter a DNS server that can resolve the name of that authentication server. As described in the Prerequisites for All Deployments, the VNet topology used for the Horizon Cloud on Microsoft Azure deployment must be able to communicate with your DNS server that you want providing external name resolution during the deployment of the Unified Access Gateway instances and for their ongoing operations. By default, the DNS server that is configured on the VNet into which the instances are deployed is the one used. When you specify addresses in <strong>DNS Addresses</strong>, the deployed Unified Access Gateway instances use those addresses in addition to the DNS server information in your VNet's configuration.</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 for communication with two-factor authentication servers. When configuring this pod to use two-factor authentication with an on-premises authentication server, you must enter the correct route the Unified Access Gateway instances can use to reach that server. For example, if your on-premises authentication 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 <code>ipv4-network-address/bits ipv4-gateway-address</code>, for example: <code>192.168.1.0/24 192.168.0.1, 192.168.2.0/24 192.168.0.2</code>.</td>
</tr>
</tbody>
</table>
### Option Description

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
</table>
| VM Model     | 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.  

**Important** In the current service release, the VM model used by these instances cannot be easily changed after the gateway configuration is deployed in your subscription. Changing the VM model after deployment requires deleting the gateway configuration and re-deploying it. If you anticipate your environment to scale to 2,000 sessions per pod, select **F8s_v2**. As stated in [VMware Horizon Cloud Service on Microsoft Azure Service Limits](#), the **A4_v2** VM model is only sufficient for proofs-of-concept (PoCs), pilots, or smaller environments where you know that you will not exceed 1,000 active sessions on the pod. |
| Certificate  | 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. |
| Blast Extreme TCP Port | Select the TCP port to use in the Blast Extreme TCP setting within the Unified Access Gateway configuration. That setting relates to the Blast Extreme via Blast Secure Gateway on Unified Access Gateway for data traffic sent by the client. Port 8443 is preferred because it is more efficient, provides better performance, and uses less resources on the Unified Access Gateway instances. For those reasons, the wizard has 8443 as the default value. The other choice, 443, is less efficient, less performant, and causes CPU congestion in the instances which can cause observable traffic delays in the end-user clients. The 443 choice should be used only if your organization has set client-side restrictions, such as your organization only permits 443 outbound.  

**Note** The UDP port used for Blast Extreme is unaffected by this setting, and is always UDP 8443. |
| Cipher Suites | Although in almost all cases the default settings are sufficient, Unified Access Gateway provides this feature for specifying the cryptographic algorithms used to encrypt communications between clients and the Unified Access Gateway appliances. At least one cipher suite must be selected from the on-screen list. The on-screen list displays the cipher suites allowed for the Horizon Cloud on Microsoft Azure deployment. |

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](#).
6  (Optional) In the **Azure Resource Tags** section, optionally add custom tags to the resource groups that contain all the internal and external Unified Access Gateway instances that you have configured for the pod.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Inherit Pod Tags</td>
<td>Switch on this toggle to add the pod’s resource tags to the resource groups containing all the Unified Access Gateway instances that you have configured. Each resource group receives the resource tags that you defined in the Pod Setup wizard step. Switch off this toggle to define new resource tags for the Unified Access Gateway instances.</td>
</tr>
</tbody>
</table>
| Azure Resource Tags| This setting becomes visible when you switch off the Inherit Pod Tags toggle. Use this setting to add new resource tags to the resource groups containing the Unified Access Gateway instances that you have configured. To create the first tag, enter information in the Name and Value fields. To create an additional tag, click + and then enter information in the Name and Value fields that appear below the existing ones.  
  - You can create a maximum of 10 tags.  
  - The tag name is limited to 512 characters, and the tag value is limited to 256 characters. For storage accounts, the tag name is limited to 128 characters, and the tag value is limited to 256 characters.  
  - Tag names cannot contain the following characters: `< > % & \ ? /  
  - Tag names cannot contain these case-insensitive strings: `azure, windows, microsoft`  
  - Tag names and tag values can only contain ASCII characters. Blank spaces and characters outside the standard 128-character ASCII set (also known as high ASCII or extended ASCII characters) are not allowed. |

**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.
When two-factor authentication details are specified in the wizard for a gateway configuration, then during the pod deployment process, the pod deployer configures the gateway configuration's corresponding deployed Unified Access Gateway appliances with the specified two-factor authentication details.

As described in the Unified Access Gateway documentation, when the Unified Access Gateway appliances are configured for two-factor authentication, the Unified Access Gateway appliances authenticate the incoming user sessions according to your specified two-factor authentication policies. After Unified Access Gateway authenticates a user session according to your specified authentication policy, then Unified Access Gateway forwards that end-user client request for a desktop or application launch to the deployed pod manager to establish a connection session between the client and an available desktop or application.

**Important** After the pod is deployed, when you plan to configure your tenant’s Universal Broker settings to use two-factor authentication, and you have deployed the pod with both an external gateway configuration and an internal gateway configuration, additional post-deployment steps might be required to ensure Universal Broker can distinguish between an external end user and an internal end user for the purposes of appropriately applying the two-factor authentication settings specified for Universal Broker. For details, see [Best Practices When Implementing Two-Factor Authentication in a Universal Broker Environment](#).

**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 are using a RADIUS authentication server and have both a primary and secondary server, obtain the information for each of them.

**RADIUS**

If you are configuring settings for both a primary and auxiliary RADIUS 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 1812/UDP for RADIUS.
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.

RSA SecurID

**Note** The RSA SecurID type is supported with Horizon Cloud on Microsoft Azure deployments that are running manifest 3139.x or later. The UI option to specify the RSA SecurID type in the Add Pod and Edit Pod wizards will become visible to select in the wizards starting in mid-March 2022.

- Access key from the RSA SecurID authentication manager server.
- RSA SecurID communication port number. Typically 5555, as set in the RSA Authentication Manager system settings for RSA SecurID Authentication API.
- Host name of the RSA SecurID authentication manager server.
- IP address of that RSA SecurID authentication manager server.
- If the RSA SecurID authentication manager server or its load balancer server has a self-signed certificate, you will need the CA certificate to provide in the Add Pod wizard. The certificate should be in PEM format (file types .cer or .cert or .pem)

**Procedure**

1. Switch on the **Enable two-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 type, **Radius** or **RSA SecurID**.

Currently, the available, supported types are RADIUS and RSA SecurID.

After selecting the type, the **Two-factor Authentication Configuration** menu automatically reflects that you are adding a configuration of that selected type. For example, if **RSA SecurID** type is selected, the **Two-factor Authentication Configuration** menu displays **New RSA SecurID**.

3 In the **Configuration 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.

The wizard displays fields based on the configuration that a Horizon Cloud on Microsoft Azure deployment supports using with its gateway configurations. The fields vary according to the selected two-factor authentication type. Refer to the table below that corresponds to your selected type, RADIUS or RSA SecurID.

**RADIUS**

As you complete the fields, specifying details about the primary authentication server is required. If you have a secondary authentication server, enable the **Auxiliary Server** toggle and specify the details for that server also.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Display Name</td>
<td>You can leave this field blank. Even though this field is visible in the wizard, it only sets an internal name in the Unified Access Gateway configuration. This name is not used by Horizon clients.</td>
</tr>
<tr>
<td>Display Hint</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: <strong>Enter your DisplayHint user name and passcode</strong>, where <strong>DisplayHint</strong> 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 <strong>Example Company user name and domain password below user name and passcode</strong> for would result in a prompt to the end user that says: <strong>Enter your Example Company user name and domain password below for user name and passcode</strong>.</td>
</tr>
<tr>
<td>Option</td>
<td>Description</td>
</tr>
<tr>
<td>---------------------</td>
<td>----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Name ID Suffix</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 will be appended to the SAML assertion user name that is sent in the request to the pod manager. For example, if the user name is entered as user1 on the login screen and a name ID suffix <a href="mailto:of@example.com">of@example.com</a> was specified here, a SAML assertion user name of <a href="mailto:user1@example.com">user1@example.com</a> is sent in the request.</td>
</tr>
<tr>
<td>Number of Iterations</td>
<td>Enter the maximum number of failed authentication attempts that a user is allowed when attempting to log in using this RADIUS system.</td>
</tr>
</tbody>
</table>
| Maintain Username    | Enable this toggle to maintain the user’s Active Directory username during the authentication flow that transpires among the client, the Unified Access Gateway instance, and the RADIUS service. When enabled:  
  - The user must have the same username credentials for RADIUS as for their Active Directory authentication.  
  - The user cannot change the username in the login screen.  
If this toggle is switched off, 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. |
| Host Name / IP Address | Enter the DNS name or the IP address of the authentication server.                                                                                                                               |
| Shared Secret        | Enter the secret for communicating with the authentication server. The value must be identical to the server-configured value.                                                                                     |
| Authentication Port  | Specify the UDP port configured on the authentication server for sending or receiving authentication traffic. The default is 1812.                                                                         |
| Accounting Port      | Optionally specify the UDP port configured on the authentication server for sending or receiving accounting traffic. The default is 1813.                                                                   |
| Mechanism            | Select the authentication protocol that is supported by the specified authentication server and which you want the deployed pod to use.                                                               |
| Server Timeout       | 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.                                      |
| Max Number of Retries | Specify the maximum number of times the pod should retry failed requests to the authentication server.                                                                                               |
| Realm Prefix         | 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. |
| Realm Suffix         | 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 user1@example.com to the authentication server. |

RSA SecurID
<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Access Key</td>
<td>Type in the access key for your RSA SecurID system, obtained in the system’s RSA SecurID Authentication API settings.</td>
</tr>
<tr>
<td>Server Port</td>
<td>Specify the value configured in your system’s RSA SecurID Authentication API settings for the communication port, typically 5555 by default.</td>
</tr>
<tr>
<td>Server Host Name</td>
<td>Enter the DNS name of the authentication server.</td>
</tr>
<tr>
<td>Server IP Address</td>
<td>Enter the IP address of the authentication server.</td>
</tr>
<tr>
<td>Number of Iterations</td>
<td>Enter the maximum number of failed authentication attempts that a user is allowed before they are locked out for one hour. The default is five (5) attempts.</td>
</tr>
<tr>
<td>CA Certificate</td>
<td>This item is required when your RSA SecurID Authentication Manager server or its load balancer uses a self-signed certificate. In this case, copy the CA certificate and paste it into this field. As described above in this page, the certificate information should be provided in PEM format. When the server has a certificate signed by a public Certificate Authority (CA), this field is optional.</td>
</tr>
<tr>
<td>Authentication</td>
<td>Specify the number of seconds you want the authentication attempt to be available between the Unified Access Gateway instances and the RSA SecurID authentication server before timing out. The default value is 180 seconds.</td>
</tr>
<tr>
<td>Timeout</td>
<td></td>
</tr>
</tbody>
</table>

**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 that describes that the endpoint must be enabled on your management subnet, you must 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**

The deployment can take between 30 to 45 minutes, depending on network traffic between Microsoft Azure Cloud and the Horizon Cloud control plane.
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** 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 traffic to host names that the deployer needs access to, as listed in the [DNS Requirements for a Horizon Cloud Pod in Microsoft and Related Service Features](#) page.

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, the likely cause is something about your environment’s network configuration or firewall that is disallowing access to one or more of the required locations listed in the [DNS Requirements for a Horizon Cloud Pod in Microsoft and Related Service Features](#) page. 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 the *.azure.com host names. You can run some tests to verify if your environment’s networking is configured properly for the pod’s requirements. See the content in the pages starting with [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>A minute or two</td>
</tr>
<tr>
<td>Downloading</td>
<td>A minute or two</td>
</tr>
<tr>
<td>Building</td>
<td>20 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.

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. See the CNAME information described in How to Obtain the Horizon Cloud Pod Gateway’s Load Balancer Information to Map in Your DNS Server.

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.

If you specified two-factor authentication for the pod’s gateway configurations, you must complete the following tasks.

- If the pod’s external gateway has two-factor authentication configured and the two-factor authentication server is not reachable within the same VNet topology into which the gateway’s Unified Access Gateway instances are deployed, configure that two-factor authentication server to allow communication from the IP address of the external gateway's load balancer.

  In this scenario where the two-factor authentication server is not reachable within the same VNet topology as the gateway deployment, the Unified Access Gateway instances attempt contact with that server using that load balancer address. To allow that communication traffic, ensure the load balancer resource's IP address that is in that external gateway's resource group is specified as a client or a registered agent in your two-factor authentication server’s configuration. Refer to the documentation for your two-factor authentication server for the specifics on how to allow that communication.

- If your two-factor authentication server is reachable within the same VNet topology, configure the two-factor authentication server to allow communication from the appropriate NICs that were created for the deployment’s Unified Access Gateway instances in Microsoft Azure.

  Your network administrator determines the two-factor authentication server’s network visibility to the Azure VNet topology and its subnets used for the deployment. The two-factor authentication server must allow communication from the IP addresses of the Unified Access Gateway instances' NICs that correspond to the subnet for which your network administrator has given network visibility to the two-factor authentication 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 and its gateways go through an update.

To support communication traffic between the gateway and the two-factor authentication server both for ongoing pod operations and after each pod update, ensure the IP addresses of those four NICs are specified as clients or as registered agents in that server’s configuration. Refer to the documentation for your two-factor authentication server for the specifics on how to allow that communication.

For information on how to obtain those IP addresses, see Update Your Two-Factor Authentication System with the Required Horizon Cloud Pod Gateway Information topic.

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 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 and Related Service Features. 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.
5. 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 can 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** Even though all of these manual tests succeed, 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, the pod deployment can get stuck in pending state. If this description matches your 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. **Horizon Cloud Pod Deployment Troubleshooting - Create an SSH Key Pair**
   As part of this troubleshooting, a test Linux VM is deployed into your Microsoft Azure subscription. To authenticate to the test Linux VM, you need an SSH key pair. You create the key pair on the system you 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.

Horizon Cloud Pod Deployment Troubleshooting - Create an SSH Key Pair
As part of this troubleshooting, a test Linux VM is deployed into your Microsoft Azure subscription. To authenticate to the test Linux VM, you need an SSH key pair. You create the key pair on the system you use to SSH connect to the test VM. This step is optional if you already have a key pair on that system.

To create this SSH key pair, you can use either a Microsoft Windows or a Linux system. The steps for both types of systems are described here. Select 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).

   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.
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 Follow the on-screen guidance to move your cursor around randomly in the blank area underneath the progress bar. As the PuTTY user interface says, moving your cursor around in the area adds required randomness into the process.

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.

   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 a Windows 10 system, the Pageant icon will be loaded into the 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.

   **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
   ~bash-4.1$ 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:

   ```text
   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 Horizon Cloud Pod Deployment Troubleshooting - 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 did not use the Add Pod wizard's option to use your own named subnets for the pod and instead entered the CIDRs for the subnets, the pod deployer creates the pod's management subnet. At the point that the deployment process failed, the process might already have created the pod’s management subnet in the VNet.

- If the deployer already has created that management subnet, 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. If you had the pod deployer automatically create the pod’s subnets (you did not use the option to use your own named subnets for the pod) the pod’s management subnet will have a name in the pattern `vmw-hcs-podID-net-management`, where `podID` is the pod’s UUID. Otherwise, the pod's management subnet is the one that you created for the pod deployment.

- If the failed 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. In the portal, create a compute VM from the Azure Marketplace and base that VM on an Ubuntu Server LTS model type.

   At the time of this writing, the Ubuntu Server 20.04 LTS was available to choose from the Azure Marketplace.

3. When creating this test Linux VM, follow the wizard UI and to configure its required options. Ensure that you configure the following items as indicated below.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Subscription</td>
<td>Match the one you selected in the Add Pod wizard 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. Follow the on-screen prompt to create a 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>Region</td>
<td>Match the one you selected in the Add Pod wizard for your pod.</td>
</tr>
<tr>
<td>Option</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Size</td>
<td>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, such as a 2 vCPU model.</td>
</tr>
<tr>
<td>Username</td>
<td>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 source</td>
<td>Select <strong>Use existing public key</strong>. The SSH public key field will appear with that selection, and you can paste in your SSH public key.</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>Public inbound ports</td>
<td>Allow the selected SSH (22) port so that you can perform the testing with this test VM.</td>
</tr>
<tr>
<td>Virtual network</td>
<td>Select the same VNet that was used for the failed pod deployment.</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. 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</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>

4. In the wizard’s final step, verify that the key pieces of information (subscription, regional location, virtual network, and subnet) match the ones that you are using for your pod, and then submit to create the 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 (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

   \text{testvm\_username@testvmip}

   substituting the test VM’s user name and IP address for \text{testvm\_username} and \text{testvmip} in the string.

   \textbf{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 this sample:

   \text{testvmadmin@40.121.180.132}
Results

When the SSH connection is established, a command-line window displays.

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:

```
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.

![Command-line window](image)

**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** Even though all of these manual tests succeed, 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, the pod deployment can get stuck in pending state. If this description matches your 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.

   ```bash
   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:

   ```bash
   testvmadmin@HCS-testingVM:~$ dig skylo.local
   ; <<>> DiG 9.10.3-P4-Ubuntu <<>> skylo.local
   ;; global options: +cmd
   ;; Got answer:
   ```
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
```
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 and Related Service Features**, 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 and Related Service Features.

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 True SSO or two-factor authentication with an authentication server, 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.

   - 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.
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.
Revision History — Changelog — Onboarding to Horizon Cloud for Microsoft Azure and Horizon Pods

This documentation topic provides the history of substantive changes to Onboarding to Horizon Cloud for Microsoft Azure and Horizon Pods — Deployment Guide.

**Note** Only substantive and significant changes made to the guide's topics starting from September 17, 2019 are provided. Details about revisions prior to that date are not available. Also, minor content revisions such as fixing typos, format changes such as turning lists into tables, and other such insignificant changes are not provided.

### August 2022

<table>
<thead>
<tr>
<th>Revision</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>30 AUG 2022</td>
<td>An issue about the enable syslog server feature is added to the Horizon Cloud - Known Issues page.</td>
</tr>
<tr>
<td>9 AUG 2022</td>
<td>Updates for the new features corresponding to the August 2022 What's New in the Horizon Cloud Release Notes.</td>
</tr>
</tbody>
</table>

### June 2022

<table>
<thead>
<tr>
<th>Revision</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>28 JUN 2022</td>
<td>Added a new required DNS name for the Horizon Edge virtual appliance to DNS Requirements for a Horizon Cloud Pod in Microsoft and Related Service Features.</td>
</tr>
<tr>
<td>26 JUN 2022</td>
<td>Updates to those documentation pages that describe logging in to the Horizon Universal Console. These updates correspond to the June 26, 2022 update in the Horizon Cloud Release Notes. The console login now incorporates authentication using VMware Cloud services.</td>
</tr>
<tr>
<td>23 JUN 2022</td>
<td>Added a new required DNS name to DNS Requirements for a Horizon Cloud Pod in Microsoft and Related Service Features.</td>
</tr>
</tbody>
</table>

### April 2022

<table>
<thead>
<tr>
<th>Revision</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>26 APR 2022</td>
<td>Updates for the new features corresponding to the April 2022 What's New in the Horizon Cloud Release Notes.</td>
</tr>
</tbody>
</table>
### March 2022

<table>
<thead>
<tr>
<th>Revision</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>9 MAR 2022</td>
<td>Updates for the new features corresponding to the March 2022 What's New in the Horizon Cloud Release Notes.</td>
</tr>
</tbody>
</table>

### February 2022

<table>
<thead>
<tr>
<th>Revision</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>8 FEB 2022</td>
<td>Added a limitation into the Horizon Cloud - Known Limitations about how the console does not prevent you from using an image obtained from origins other than the Azure Marketplace, even though to be supported for use in Horizon Cloud on Microsoft Azure, all imported base images must be built from Windows-based VMs that are sourced from the Azure Marketplace.</td>
</tr>
<tr>
<td>3 FEB 2022</td>
<td>Updates for the new features corresponding to the February 2022 What's New in the Horizon Cloud Release Notes.</td>
</tr>
</tbody>
</table>

### November 30, 2021

<table>
<thead>
<tr>
<th>Revision</th>
<th>Description</th>
</tr>
</thead>
</table>

### October 12, 2021

<table>
<thead>
<tr>
<th>Revision</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>12 OCT 2021</td>
<td>Updates for the new features corresponding to the October 12, 2021 What's New in the Horizon Cloud Release Notes.</td>
</tr>
</tbody>
</table>

### September 7, 2021

<table>
<thead>
<tr>
<th>Revision</th>
<th>Description</th>
</tr>
</thead>
</table>

### August 18, 2021

<table>
<thead>
<tr>
<th>Revision</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>18 AUG 2021</td>
<td>Updates in the topic about Create a Horizon Cloud App Registration in the Pod's Subscription to align with the new behavior in the Microsoft Azure portal that limits the duration of the secret key's expiration to no greater than two (2) years.</td>
</tr>
</tbody>
</table>
### August 10, 2021

<table>
<thead>
<tr>
<th>Revision</th>
<th>Description</th>
</tr>
</thead>
</table>

### July 15, 2021

<table>
<thead>
<tr>
<th>Revision</th>
<th>Description</th>
</tr>
</thead>
</table>

### June 29, 2021

<table>
<thead>
<tr>
<th>Revision</th>
<th>Description</th>
</tr>
</thead>
</table>
| 29 JUN 2021  | Updated the following topics to add guidance about which VM models to select for the Unified Access Gateway appliances in a pod's external and internal gateway configuration.  
- VMware Horizon Cloud Service on Microsoft Azure Service Limits  
- Chapter 3 VMware Horizon Cloud Service on Microsoft Azure Requirements Checklist For New Pod Deployments Starting From the v2207 Service Release  
- Specify the Horizon Cloud Pod's Gateway Configuration |

### June 9, 2021

<table>
<thead>
<tr>
<th>Revision</th>
<th>Description</th>
</tr>
</thead>
</table>
| 9 JUN 2021   | Updated the following topics to add information about the Europe-3 (Germany) regional control plane instance and to remove mention of the DNS names in the pattern query-prod*. Reachability to the query-prod* DNS names is not required.  
- DNS Requirements for a Horizon Cloud Pod in Microsoft and Related Service Features  
- DNS, Ports, and Protocols Requirements When Using Horizon Cloud Connector and a Horizon Pod  
Also, a correction is made to the DNS Requirements for a Horizon Cloud Pod in Microsoft and Related Service Features topic. The source subnet for the table row about the Cloud Management Service’s DNS names in the pattern kinesis.* is the pod’s management subnet. Previously, that table row listed the tenant subnet as the source subnet. |

### May 20, 2021

<table>
<thead>
<tr>
<th>Revision</th>
<th>Description</th>
</tr>
</thead>
</table>
### April 14, 2021

<table>
<thead>
<tr>
<th>Revision</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>14 APR 2021</td>
<td>Updates for the new features corresponding to the April 14, 2021 What’s New in the Horizon Cloud Release Notes.</td>
</tr>
</tbody>
</table>

### March 25, 2021

<table>
<thead>
<tr>
<th>Revision</th>
<th>Description</th>
</tr>
</thead>
</table>

### March 9, 2021

<table>
<thead>
<tr>
<th>Revision</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>9 MAR 2021</td>
<td>Updates for the new features corresponding to the March 9, 2021 What’s New in the Horizon Cloud Release Notes.</td>
</tr>
</tbody>
</table>

### March 1, 2021

<table>
<thead>
<tr>
<th>Revision</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 MAR 2021</td>
<td>Updates related to the removal of the requirement that the domain join account must be in an Active Directory group that has the Super Administrator role. This requirement is only when your pod fleet has Horizon Cloud pods in Microsoft Azure that are running a manifest older than 1600.0. For additional details, see Service Accounts That Horizon Cloud Requires for Its Operations.</td>
</tr>
</tbody>
</table>

### January 7, 2021

<table>
<thead>
<tr>
<th>Revision</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>7 JAN 2021</td>
<td>Updates for the new features corresponding to the January 7, 2021 What’s New in the Horizon Cloud Release Notes.</td>
</tr>
</tbody>
</table>

### December 15, 2020

<table>
<thead>
<tr>
<th>Revision</th>
<th>Description</th>
</tr>
</thead>
</table>
### December 2, 2020

<table>
<thead>
<tr>
<th>Revision</th>
<th>Description</th>
</tr>
</thead>
</table>

### November 24, 2020

<table>
<thead>
<tr>
<th>Revision</th>
<th>Description</th>
</tr>
</thead>
</table>

### November 4, 2020

<table>
<thead>
<tr>
<th>Revision</th>
<th>Description</th>
</tr>
</thead>
</table>

### October 8, 2020

<table>
<thead>
<tr>
<th>Revision</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>8 OCT 2020</td>
<td>Updates for the new features corresponding to the October 8, 2020 What's New in the Horizon Cloud Release Notes.</td>
</tr>
</tbody>
</table>

### July 9, 2020 — October 7, 2020

<table>
<thead>
<tr>
<th>Revision</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>9 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 and Related Service Features and DNS, Ports, and Protocols Requirements When Using Horizon Cloud Connector and a Horizon Pod.</td>
</tr>
<tr>
<td>13 JUL 2020</td>
<td>Pods that can have the high availability (HA) feature are now supported for Microsoft Azure Government (US Gov Virginia, US Gov Arizona, US Gov Texas).</td>
</tr>
<tr>
<td>9 JUL 2020</td>
<td>Updates for the new features corresponding to the July 9, 2020 What's New in the Horizon Cloud Release Notes.</td>
</tr>
</tbody>
</table>
## March 17, 2020 — July 8, 2020

<table>
<thead>
<tr>
<th>Revision</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>9 JUN 2020</td>
<td>Updates made to align this guide with the June 9, 2020 What’s New in the <a href="#">Horizon Cloud Release Notes</a>. The regional names that appear in the Welcome email have been updated to use human-friendly names. The following doc topic is also updated to align with that change: DNS Requirements for a Horizon Cloud Pod in Microsoft and Related Service Features</td>
</tr>
<tr>
<td>27 MAY 2020</td>
<td>Updates made to align this guide with the May 27, 2020 What’s New in the <a href="#">Horizon Cloud Release Notes</a>.</td>
</tr>
<tr>
<td>12 MAY 2020</td>
<td>Updates made to align this guide with the May 12, 2020 What’s New in the <a href="#">Horizon Cloud Release Notes</a>. Also corrected entries in the table for agent-related ports and protocols requirements in doc topics Horizon Cloud Pod - Ports and Protocols Requirements and Chapter 9 Deprecated - Manifests 1493.1 or Earlier - Horizon Cloud Pod - Ports and Protocols.</td>
</tr>
<tr>
<td>14 APR 2020</td>
<td>Updates made to align this guide with the April 13, 2020 What’s New in the <a href="#">Horizon Cloud Release Notes</a>.</td>
</tr>
<tr>
<td>17 MAR 2020</td>
<td>Updates for the new features corresponding to the March 17, 2020 What’s New in the <a href="#">Horizon Cloud Release Notes</a>.</td>
</tr>
</tbody>
</table>

## December 13, 2019 — March 16, 2020

<table>
<thead>
<tr>
<th>Revision</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>25 FEB 2020</td>
<td>Updated the following topics for the stated changes:</td>
</tr>
<tr>
<td></td>
<td>- <strong>Added</strong> Microsoft.Network/virtualNetworks/virtualNetworkPeerings/read to the list in <a href="#">When Your Organization Prefers to Use a Custom Role for the Horizon Cloud App Registration</a>.</td>
</tr>
<tr>
<td></td>
<td>- Added resource provider Microsoft.Sql to the list in <a href="#">Create a Horizon Cloud App Registration in the Pod’s Subscription</a>.</td>
</tr>
<tr>
<td></td>
<td>- Added a row for the jumpbox VM and pod manager VM to reach the Unified Access Gateway VMs using port 9443/TCP to <a href="#">Horizon Cloud Pod - Ports and Protocols Requirements</a>. This port is required to configure settings in the Unified Access Gateway during pod deployment and whenever editing the pod to change its Unified Access Gateway settings.</td>
</tr>
<tr>
<td></td>
<td>- Corrected port 443 to 8433 as the port requirement for traffic from the pod’s Unified Access Gateway VMs to the pod’s Microsoft Azure load balancer for login authentication traffic, in <a href="#">Horizon Cloud Pod - Ports and Protocols Requirements</a>.</td>
</tr>
<tr>
<td>13 JAN 2020</td>
<td>Updated proxy-related information for the Horizon 7 Cloud Connector. Updated topics include <a href="#">Onboarding a Horizon Pod to Horizon Cloud Control Plane</a> and <a href="#">Horizon Pod and Horizon Cloud Connector - Preparing to Onboard to Control Plane Services</a>.</td>
</tr>
<tr>
<td>6 JAN 2020</td>
<td>New information added to topics about <a href="#">Enable SSH Access to the Horizon Cloud Connector Using a Command Line Interface</a>, <a href="#">Chapter 4 VMware Horizon Pods with Horizon Cloud Control Plane - Requirements Checklist - Appropriately Updated Starting from the v2204 Service Release</a>.</td>
</tr>
<tr>
<td>13 DEC 2019</td>
<td>Updates for the new features corresponding to the December 13, 2020 What’s New in the <a href="#">Horizon Cloud Release Notes</a>.</td>
</tr>
</tbody>
</table>
### September 17, 2019 — December 12, 2019

<table>
<thead>
<tr>
<th>Revision</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>21 NOV 2019</td>
<td>Updated the resource providers list with additional ones needed for a subscription in Microsoft Azure related to the latest pod architecture, in Step 8 of documentation topic <a href="http://example.com">Create a Horizon Cloud App Registration in the Pod’s Subscription</a>.</td>
</tr>
<tr>
<td>17 SEP 2019</td>
<td>Updates for the new features corresponding to the September 17, 2020 What’s New in the <a href="http://example.com">Horizon Cloud Release Notes</a>.</td>
</tr>
</tbody>
</table>
Horizon Cloud on Microsoft Azure deployments using manifests 1493.1 or earlier have reached end of general support.

Therefore, this documentation page about ports and protocols for such pods is no longer relevant. This deprecated page will remain for a short time until an automatic redirect to the page for supported manifests is put into place.

- For the Horizon Cloud on Microsoft Azure pod manifest support statements, see VMware KB 86476.

- For the ports and protocols information for supported manifests, see Horizon Cloud Pod - Ports and Protocols Requirements.