Horizon Cloud Deployment Guide

For service level starting September 17, 2019
VMware Horizon Cloud Service
You can find the most up-to-date technical documentation on the VMware website at:
https://docs.vmware.com/

If you have comments about this documentation, submit your feedback to
docfeedback@vmware.com
Contents

About This Horizon Cloud Deployment Guide  4

1 Introduction to Horizon Cloud and Cloud-Connected Pods  7

2 Deploying Your Very First Cloud-Connected Pod for Horizon Cloud  9
   When You Choose On-Premises Capacity for Your Very First Pod Deployment  10
      End-to-End Workflow When Your Very First Cloud-Connected Pod is from Connecting Horizon Cloud
      with an Existing Manually Deployed Horizon 7 Pod  12
   Connect Horizon Cloud with an Existing Manually Deployed Horizon 7 Pod  15
   Static IP and Proxy Configuration for the Horizon 7 Cloud Connector Virtual Appliance  20
   Configure a CA-Signed Certificate for the Horizon 7 Cloud Connector Virtual Appliance  21
   Horizon 7 Cloud Connector Known Considerations  24
   Set a Password Expiry Policy for the Horizon 7 Cloud Connector Root User  25
   When You Choose Microsoft Azure Cloud Capacity for Your Very First Pod Deployment  25
      Suggested Workflow for When Your Very First Cloud-Connected Pod is from Deploying into Microsoft
      Azure  30

3 Now That Your Very First Pod Is Completely Deployed and Connected to Horizon Cloud  140
About This Horizon Cloud Deployment Guide

This Horizon Cloud Deployment Guide document describes the process of deploying your very first ever cloud-connected pod for your VMware Horizon® Cloud Service™ environment. Use the topics in this guide when your environment has no pods at all, and your view of your environment is the initial one, as shown in the following screenshot. When you already have one cloud-connected pod, use the companion Horizon Cloud Administration Guide for information on deploying subsequent pods after your very first one.

If your first pod deployment is hosting a pod in Microsoft Azure, before starting the steps described in this document, see also the VMware Horizon Cloud Service on Microsoft Azure Requirements Checklist. That document is available in PDF format, and it describes various prerequisite elements needed to have your first deployment into Microsoft Azure be successful.

If your first pod deployment is cloud-connecting an existing Horizon 7 pod, before starting the steps described in this document, see also the VMware Horizon 7 with VMware Horizon Cloud Service Requirements Checklist. That document is available in PDF format, and it describes various prerequisite elements needed to have your first deployment into VMware Cloud be successful.
When your very first pod is deployed into your Horizon Cloud environment, you must perform some initial getting-started tasks before you can leverage the full power of that pod or deploy additional pods. To read about those tasks, see the companion *Horizon Cloud Administration Guide* beginning with the topics *Introduction to Horizon Cloud* and *Getting Started Using Your Horizon Cloud Environment* in that guide.

**Document Revision History**

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

<table>
<thead>
<tr>
<th>Revision</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>17 SEP 2019</td>
<td>Initial version, when the described features were deployed live into production.</td>
</tr>
<tr>
<td></td>
<td>This document describes the latest service features.</td>
</tr>
<tr>
<td></td>
<td>For pods using Microsoft Azure capacity, the latest features correspond to pod manifest version 1600 and later.</td>
</tr>
<tr>
<td></td>
<td>For cloud-connected Horizon 7 pods, the latest features correspond to Horizon 7 components at version 7.10 and later, and Horizon 7 Cloud Connector at version 1.4 and later.</td>
</tr>
</tbody>
</table>

**Intended Audience**

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

- VMware Horizon
- VMware Horizon® 7 Cloud Connector™
- VMware Unified Access Gateway™
- VMware Identity Manager™
- Virtualization technology
- Networking
- VMware Cloud™ on AWS (VMware Cloud)
- VMware Horizon® 7 on VMware Cloud™ on AWS
- Microsoft Azure

**About the Screenshots Used in This Document**

The screenshots typically:

- Show only that portion of the overall user interface screen that corresponds to the text at which point the screenshot appears, and not necessarily the full user interface.
Horizon Cloud Community

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

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

Contacting VMware Support

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

- You can submit a support request to VMware Support online using your My VMware® account or by phone.
- KB 2144012 Customer Support Guidelines provides details for getting support depending on the issue encountered.
- In the Administration Console, clicking > Support, displays the link to that KB 2144012 also.

VMware Technical Publications Glossary

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

Your overall Horizon Cloud environment consists of the VMware-hosted cloud service, your provided capacity, and VMware software deployed into that capacity and connected to the cloud service. When the VMware software installed in that capacity is appropriately configured and connected to the cloud service, that configured entity is called a cloud-connected pod. You can then work with all of your cloud-connected pods and the available desktop-as-a-service features using the Horizon Cloud Administration Console.

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

**cloud-connected pod**
VMware software deployed into a supported capacity environment and connected to the cloud control plane. Supported capacity environments are ones such as Microsoft Azure cloud or VMware Cloud™ on AWS or on-premises infrastructure.

Depending on the type of capacity you are using, you can use the Horizon Cloud Administration Console for an automated pod deployment and connection to Horizon Cloud. For some types of pods, even though they cannot be automatically deployed and configured, you can still connect Horizon Cloud to those pods that already exist. Then you can use the Administration Console with the cloud-connected pods after they are connected.

**Horizon Cloud Control Plane**
VMware hosts the Horizon Cloud control plane in the cloud. This cloud service enables the central orchestration and management of virtual desktops, remote desktop sessions, and remote applications for your users. The cloud service also manages your pods. The pods are physically located in your provided capacity environments. When you log in to the cloud service, you can see all of your pods and perform management activities across them, regardless of where they are physically located.

VMware is responsible for hosting the service and providing feature updates and enhancements for a software-as-a-service experience.
The cloud control plane also hosts a common management user interface called the Horizon Cloud Administration Console, or Administration Console for short. The Administration Console runs in industry-standard browsers. It provides IT administrators with a single location for management tasks involving user assignments and the virtual desktops, remote desktop sessions, and applications. The Administration Console is accessible from anywhere at any time, providing maximum flexibility.

**Important** The Administration 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 Administration 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. The Administration Console dynamically reflects the elements related to such features only when your license includes use of such features. For examples, see the Tour of the Horizon Cloud Administration Console topic in the *Horizon Cloud Administration Guide*.

When you are expecting to see a feature in the Administration Console and do not see it, contact your VMware account representative to verify whether your license entitles your use of that feature.

### Pod Types You Can Connect to Horizon Cloud

This Horizon Cloud release provides for the following deployment types.

**Note** To connect a pod to Horizon Cloud or use the Administration Console for an automated deployment, your customer account must have the appropriate licensing. For licensing information, contact your VMware account representative.

<table>
<thead>
<tr>
<th>Table 1-1. Pod Deployment Types</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Deployment Type</strong></td>
</tr>
<tr>
<td>VMware Horizon 7 pod located in your on-premises infrastructure</td>
</tr>
<tr>
<td>VMware Horizon 7 pod that you manually installed and configured in your VMware Cloud on AWS SDDC</td>
</tr>
<tr>
<td>Horizon Cloud pod deployed by Horizon Cloud into your Microsoft Azure cloud capacity</td>
</tr>
</tbody>
</table>

**Important** For production environments, ensure the VM models used for your farms and desktop assignments have a minimum of two (2) CPUs. VMware scale testing has shown that using 2 CPUs or more avoids unexpected end-user connection issues. Even though the system does not prevent you from choosing a VM model with a single CPU, you should use such VM models for tests or proof-of-concepts only.
Deploying Your Very First Cloud-Connected Pod for Horizon Cloud

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

The following screenshot is a representative example of what a brand new fresh Horizon Cloud environment looks like when you log in to your account for the first time.

![Horizon Cloud Getting Started](image)

That first-time screen is oriented around the idea of adding capacity. You can think of adding capacity here as equivalent to deploying pods in various capacity environments and connecting those pods to your overall Horizon Cloud environment.
### What the on-screen text says | How that text relates to your first cloud-connected pod
--- | ---
Learn how to add On-Premises capacity... | Adding on-premises capacity from this point in time makes your first cloud-connected pod an existing Horizon 7 pod that you have on premises or that you manually configured in VMware Cloud. If you are interested in this type for your first cloud-connected pod, read how best to proceed from the topic **End-to-End Workflow When Your Very First Cloud-Connected Pod is from Connecting Horizon Cloud with an Existing Manually Deployed Horizon 7 Pod.**

Add Cloud Capacity ... Microsoft Azure Cloud | Adding Microsoft Azure cloud capacity from this point in time results in a wizard-driven, automatic deployment of a pod into Microsoft Azure. When that process is completed, that pod is your first cloud-connected pod. If you are interested in this type for your first cloud-connected pod, learn how to proceed from the topic **When You Choose Microsoft Azure Cloud Capacity for Your Very First Pod Deployment** and its subtopics.

---

After your Horizon Cloud environment gets its first cloud-connected pod, you then use the companion **Horizon Cloud Administration Guide** to continue the Active Directory domain registration steps with that first cloud-connected pod. After the domain registration is completed, you continue using that companion **Horizon Cloud Administration Guide** to learn about all of the workflows you can do with Horizon Cloud, including adding additional pods. In that guide, begin with the **Introduction to Horizon Cloud** topic, the **Getting Started Using Your Horizon Cloud Environment**, and their subtopics.

**Note**
- If you have a manually created Horizon 7 pod in VMware Cloud that you’d like to connect to Horizon Cloud, you can use the add on-premises capacity workflow to connect that existing pod. If that is the first add-capacity workflow you complete, then that pod would become your first cloud-connected pod. Before starting the steps described in this document, see also the **VMware Horizon 7 with VMware Horizon Cloud Service Requirements Checklist**. That document is available in PDF format, and it gives a nice overview of the various prerequisite elements needed to have success with your first pod deployment into VMware Cloud.

- If you are adding Microsoft Azure cloud capacity as your first cloud-connected pod, before starting the steps described in this document, see also the **VMware Horizon Cloud Service on Microsoft Azure Requirements Checklist**. That document is available in PDF format, and it gives a nice overview of the various prerequisite elements needed to have success with your first pod deployment into Microsoft Azure.

This chapter includes the following topics:

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

### When You Choose On-Premises Capacity for Your Very First Pod Deployment

Here, on-premises capacity means your VMware Horizon 7 deployments that you manually installed and configured in your on-premises infrastructure or in your VMware Cloud SDDC. You connect such Horizon
7 deployments to Horizon Cloud for two primary use cases: to activate a subscription license for your Horizon 7 deployment and to use cloud-hosted services with it. Activating the subscription license provides for leveraging those cloud-hosted services. Such services include features and workflows for your cloud-connected pods. Both use cases require use of the VMware Horizon 7 Cloud Connector, a component that connects your Horizon 7 deployment with the cloud-based management plane provided by Horizon Cloud. That connection provides both for pushing the subscription license to your Horizon 7 deployment and for enabling the cloud-hosted services.

**Important** Your Horizon Cloud environment can consist of pods in Microsoft Azure and Horizon 7 pods on-premises and in VMware Cloud. As a result, all of those cloud-connected pods must have line of sight to the same set of Active Directory domains. If your environment already has pods in Microsoft Azure and you are connecting your first Horizon 7 pod, you must ensure the Horizon 7 pod will be able to have line of sight to the Active Directory domains that are already registered with your Horizon Cloud environment at the time you connect the Horizon 7 pod. See the Active Directory-related topics in the Horizon Cloud Administration Guide for more details.

---

**Activating Subscription Licenses and Enabling Cloud-Hosted Services for Horizon 7 Pods Using the VMware Horizon 7 Cloud Connector**

The Horizon 7 Cloud Connector is a virtual appliance that bridges a Horizon 7 pod with Horizon Cloud. Horizon 7 Cloud Connector is required for using cloud-hosted services with your Horizon 7 pod including Horizon 7 subscription licenses, health status dashboard, and Horizon Help Desk Tool.

**Horizon 7 subscription licenses**

Horizon 7 subscription licenses are available with the standalone Horizon 7 package and as part of the Workspace ONE Enterprise bundle. The Horizon 7 subscription license provides the same product with more flexible deployment options. Horizon 7 subscription licenses enable Horizon 7 deployment in the data center, private cloud, and in VMware Cloud on AWS. When you have completed the steps using the Horizon 7 Cloud Connector to connect your pod to Horizon Cloud, you can apply the subscription license using the Horizon Administration Console or the Horizon Console. For the steps to apply the subscription license after connecting your pod to Horizon Cloud, see *Horizon Console Administration* in the *Horizon 7 documentation*.

You must have an active My VMware account to purchase a Horizon 7 license from [https://my.vmware.com](https://my.vmware.com). You then receive a subscription email with the link to download the Horizon 7 Cloud Connector as an OVA file.

When you deploy the Horizon 7 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 7 Cloud Connector virtual appliance connects the Connection Server to Horizon Cloud to manage the Horizon 7 subscription license and
other services. With a Horizon 7 subscription license, you do not need to manually enter a Horizon 7 license key for the VMware Horizon 7 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 7 pods

When you have completed the steps using the Horizon 7 Cloud Connector to connect your pod to Horizon Cloud, in addition to applying a subscription license, you can leverage those cloud-hosted services, features, and workflows that are provided by Horizon Cloud and which are available to you according to that subscription license. Such services include features and workflows for your cloud-connected pods. For information about those cloud-hosted services, see Chapter 1 Introduction to Horizon Cloud and Cloud-Connected Pods.

End-to-End Workflow When Your Very First Cloud-Connected Pod is from Connecting Horizon Cloud with an Existing Manually Deployed Horizon 7 Pod

This list is a high-level of the steps when you are making your very first cloud-connected pod by connecting to an existing manually deployed Horizon pod. A manually deployed pod is one that you manually installed and configured using either on-premises capacity or using VMware Cloud capacity. After these workflow steps are completed for your very first cloud-connected pod, you can use the Horizon Cloud Administration Console features for visibility, health monitoring, and help desk services with that pod. At that point, you can also deploy additional pods.

Before beginning this workflow, you must have already installed and configured your Horizon 7 pod. For information about manually installing a Horizon 7 pod that you can use with this Horizon Cloud release:

- For manually installing pods using on-premises capacity, see the relevant installation information for the most recent version of Horizon 7 from the Horizon 7 Documentation page.
- For manually installing pods using VMware Cloud capacity, see the best practices guide for deploying Horizon 7 on VMware Cloud, available from the Horizon 7 on VMware Cloud on AWS product page.
This Horizon Cloud release provides unified visibility, health monitoring, and help desk services in the Horizon Cloud Administration Console. Perform the following steps when you are deploying your very first cloud-connected pod and you are connecting to a manually deployed pod with which you want to use the Horizon Cloud features.

**Caution** Complete all of the steps below to fully connect your first pod to Horizon Cloud before you start deploying the Horizon Cloud Connector with any subsequent manually installed pod you want to connect. Due to a known issue in this release, if you finish connecting more than one pod to the cloud using the Horizon Cloud Connector before you complete the Active Directory domain registration and Super Administrators role assignment step at least once, the Active Directory domain registration step will fail. At that point, you will have to unplug all but one of your cloud-connected Horizon 7 pods before you can successfully complete the required Active Directory domain registration and Super Administrators role assignment step.

1. Fulfill the prerequisites. See [Connect Horizon Cloud with an Existing Manually Deployed Horizon 7 Pod](#).
2. Verify you meet the DNS, ports, and protocol requirements for connecting a Horizon 7 pod with Horizon Cloud. See [DNS, Ports, and Protocols Requirements When Using Horizon 7 Cloud Connector and a Horizon 7 Pod](#).
3. Deploy the Horizon Cloud Connector into that pod environment. See [Connect Horizon Cloud with an Existing Manually Deployed Horizon 7 Pod](#).
4. Optionally configure a CA-signed certificate for the Horizon 7 Cloud Connector virtual appliance. See [Configure a CA-Signed Certificate for the Horizon 7 Cloud Connector Virtual Appliance](#).
5. Register the pod's Active Directory domain with Horizon Cloud, which includes providing names of domain-bind and domain-join accounts. See the [Horizon Cloud Administration Guide](#).
6. Give the Horizon Cloud Super Administrators role to an Active Directory group that includes that domain-join account as a member. See the [Horizon Cloud Administration Guide](#).

You can find in-depth details on how to accomplish each workflow step in the topics that are linked from each step above or in the companion guide. See the [Horizon Cloud Administration Guide](#).

### DNS, Ports, and Protocols Requirements When Using Horizon 7 Cloud Connector and a Horizon 7 Pod

When you are using the Horizon 7 Cloud Connector component with your Horizon 7 pod, you must configure your firewalls to allow the Cloud Connector to access the Domain Name Service (DNS) addresses it needs. In addition, your DNS must resolve specific names as described in this topic. Then, the Cloud Connector component 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 Cloud Connector.

As described in [When You Choose On-Premises Capacity for Your Very First Pod Deployment](#), the Cloud Connector component is used with Horizon 7 deployments to activate subscription licenses on Horizon 7 and enable use of cloud-hosted services with your Horizon 7 deployments.
Connectivity and DNS Requirements

The Connect Horizon Cloud with an Existing Manually Deployed Horizon 7 Pod include the step to use a browser to navigate to the Cloud Connector appliance’s IP address and a login screen will appear. To see that login screen requires Internet connectivity between the 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 web-socket, using outbound Internet port 443. For ongoing operations, the connection between the Cloud Connector appliance and Horizon Cloud requires that outbound Internet connection using port 443 open all the time. You must ensure the following DNS names are resolvable and reachable using the specific ports and protocols as listed in the following table.

<table>
<thead>
<tr>
<th>Source</th>
<th>Destination (DNS name)</th>
<th>Port</th>
<th>Protocol</th>
<th>Purpose</th>
</tr>
</thead>
</table>
| Horizon 7 Cloud Connector  | One of the following, depending on which Horizon Cloud control plane is specified in your Horizon Cloud account:  
  - cloud.horizon.vmware.com  
  - cloud-eu-central-1.horizon.vmware.com  
  - cloud-ap-southeast-2.horizon.vmware.com | 443  | TCP      | Horizon Cloud control plane.  
  Used for the Horizon 7 Cloud Connector to connect with the control plane.  
  - cloud.horizon.vmware.com is in the United States  
  - cloud-eu-central-1.horizon.vmware.com is in Europe  
  - cloud-ap-southeast-2.horizon.vmware.com is in Australia | Horizon Cloud control plane.  
  Used for the Horizon 7 Cloud Connector to connect with the control plane.  
  - cloud.horizon.vmware.com is in the United States  
  - cloud-eu-central-1.horizon.vmware.com is in Europe  
  - cloud-ap-southeast-2.horizon.vmware.com is in Australia |
| Horizon 7 Cloud Connector  | Depending on which regional Horizon Cloud control plane is specified in your Horizon Cloud account:  
  North America:  
  - kinesis.us-east-1.amazonaws.com  
  - query-prod-us-east-1.cms.vmware.com  
  Europe:  
  - kinesis.eu-central-1.amazonaws.com  
  - query-prod-eu-central-1.cms.vmware.com  
  Australia:  
  - kinesis.ap-southeast-2.amazonaws.com  

Ports and Protocols Required by Horizon 7 Cloud Connector Appliance

For ongoing operations between Horizon 7 Cloud Connector and Horizon Cloud, the ports and protocols in the following table are required.
Table 2-1. Horizon 7 Cloud Connector Ports

<table>
<thead>
<tr>
<th>Source</th>
<th>Target</th>
<th>Port</th>
<th>Protocol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Horizon 7 Cloud Connector</td>
<td>Horizon Cloud</td>
<td>443</td>
<td>HTTPS</td>
<td>Used to pair the Horizon 7 Cloud Connector with Horizon Cloud and transfer data.</td>
</tr>
<tr>
<td></td>
<td>Connection Server</td>
<td>443</td>
<td>HTTPS</td>
<td>API calls to Connection Server.</td>
</tr>
<tr>
<td></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 7 Cloud Connector appliance</td>
<td>Existing version of the Horizon 7 Cloud Connector appliance</td>
<td>22</td>
<td>SSH</td>
<td>Listen for requests to start the upgrade process.</td>
</tr>
<tr>
<td>Web browser</td>
<td>Horizon 7 Cloud Connector</td>
<td>443</td>
<td>HTTPS</td>
<td>Listen for the initiation of the pairing process.</td>
</tr>
<tr>
<td></td>
<td>Certificate Authority</td>
<td>*</td>
<td>HTTP, HTTPS</td>
<td>CRL or OCSP queries</td>
</tr>
<tr>
<td>Cloud Monitoring Service agent in the desktop or server VMs that are from the cloud-connected Horizon 7 on your network</td>
<td>Horizon 7 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 Cloud Connector</td>
</tr>
</tbody>
</table>

Connect Horizon Cloud with an Existing Manually Deployed Horizon 7 Pod

Connecting Horizon Cloud with your manually deployed pod is a multi-step process. After you purchase a VMware Horizon 7 subscription license, you will receive a license subscription email, which includes the link to download the Horizon 7 Cloud Connector virtual appliance. Then you install the virtual appliance and pair it with a Connection Server in the pod that you want to connect to Horizon Cloud. As a result, the pod is connected to the cloud control plane.

Download the Horizon 7 Cloud Connector from its location in VMware.com if you have not already done so using the link in your subscription email.
You might have best results deploying the appliance into your pod when you use the VMware vSphere® Web Client, which is a Flex-based user interface. If you deploy using the VMware vSphere® Client™, which is the newest HTML5-based user interface, you might encounter an error message about an invalid value. That issue is caused by a known issue in the vSphere Client, and is not an issue with the Horizon 7 Cloud Connector appliance package. If you encounter that error, deploy the appliance using the Flex-based user interface instead.

**Note** You use these steps to connect Horizon 7 environments that already exist, installed on-premises or installed in VMware Cloud™ on AWS.

**Note** Proxy SSL configuration is not supported during the deployment of the Horizon 7 Cloud Connector virtual appliance.

### Prerequisites

To connect an existing Horizon pod to the Horizon Cloud Administration Console in this release, you must meet the following prerequisites.

- Even though you can cloud-connect a Horizon 7 pod at version 7.6 and later, to obtain the most up-to-date features, the Horizon pod should be at Horizon 7 version 7.10 and later.

- You must have a My VMware account on https://my.vmware.com. This account is required to purchase a Horizon 7 subscription license and for logging in to the Horizon Cloud Administration Console.

- You must obtain a Horizon 7 subscription license.

- Use the link in the subscription license email that you receive after purchasing the subscription license to download the Horizon 7 Cloud Connector virtual appliance from my.vmware.com.

**Important** To use the most up-to-date Horizon Cloud features with the pod, the Horizon 7 Cloud Connector must be at version 1.4 or later. If your pod is already using a Horizon 7 Cloud Connector with a version prior to 1.4, you must upgrade it according to the steps in the Horizon Cloud Administration Guide.

- Verify the Connection Server with which you want to pair the Horizon 7 Cloud Connector virtual appliance. You can pair the Horizon 7 Cloud Connector virtual appliance with only one of the pod’s installed Connection Server instances at a time.

- If the Horizon 7 Cloud Connector virtual appliance is not part of the Active Directory domain to which your chosen Connection Server is joined, add the fully qualified domain name (FQDN) of the Connection Server that you want to pair with Horizon 7 Cloud Connector to the /etc/hosts file on the Horizon 7 Cloud Connector virtual appliance.

- Add the FQDN of the vCenter Server to the /etc/hosts file on the Horizon 7 Cloud Connector virtual appliance.

- If you are using the Microsoft Internet Explorer Web browser, verify that the compatibility mode is off. This setting enables viewing the Horizon 7 Cloud Connector appliance user interface in that web browser.
Deploy and join the Horizon 7 Cloud Connector virtual appliance with a static IP address to Active Directory.

**Note** Do not use IPv6 with the Horizon 7 Cloud Connector virtual appliance. IPv6 is not supported.

Obtain the DNS address, gateway address, and subnet mask for the Horizon 7 Cloud Connector virtual appliance.

If you want to use a fully qualified domain name (FQDN) for the Horizon 7 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 7 Cloud Connector virtual appliance's static IP.

Verify you meet the DNS, ports, and protocol requirements described in DNS, Ports, and Protocols Requirements When Using Horizon 7 Cloud Connector and a Horizon 7 Pod.

**Important** If you already have cloud-connected pods in your Horizon Cloud environment to which you are connecting this Horizon 7 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 Horizon 7 pod, you must ensure that the Horizon 7 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 7 pod, you must ensure:

- The Horizon 7 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 7 pod's Active Directory domain, the domain you are using in the following steps to pair the Horizon 7 Cloud Connector virtual appliance with the Horizon 7 pod's Connection Server.

**Procedure**

1. Download the Horizon 7 Cloud Connector appliance from the link provided in the subscription email you received.

   The Horizon 7 Cloud Connector appliance is available as an OVA file.

   **Important** Ensure that the downloaded version is version 1.4 or later, to enable the most up-to-date features.
2 Use vSphere Web Client to deploy the Horizon 7 Cloud Connector appliance as an OVF template into your Horizon pod.

For general information about deploying OVF templates, see the *vSphere Virtual Machine Administration* documentation.

**Note** Keep in mind the following items:

- When you enter a root password for the OVF template, you must verify that the password contains a minimum of eight characters with one capital, one numeric, and one special character.
- If you attempt to deploy the appliance using the vSphere Client (an HTML5-based user interface) and you encounter an error message about an invalid value, switch to deploying the appliance using the vSphere Web Client (the Flex-based user interface) instead.

3 In vSphere Web Client, power on the Horizon 7 Cloud Connector appliance.

The Horizon 7 Cloud Connector appliance user interface IP address appears.

4 Using a browser, navigate to the Horizon 7 Cloud Connector appliance’s IP address.

The login screen appears.

5 Log in to the Horizon 7 Cloud Connector user interface using your My VMware account credentials.

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

The first step of the setup wizard appears.

6 In the **Connect to Horizon 7 Connection Server** box, enter the FQDN of that Connection Server instance.

When the box contains an FQDN, the **Connect** button appears.

7 Click **Connect**.

The Horizon 7 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.

8 If the Connection Server does not have a valid Root CA certificate, click the check box to verify that the displayed certificate information is accurate and 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.

9 Type the domain name, user name, and password used by the Connection Server and click **Connect**.

**Note** At this point, the system detects whether the specified Connection Server instance is already paired with another instance of the Horizon 7 Cloud Connector. In this case, the page displays a message and an **Accept** button to delete the existing pairing and pair this Connection Server with the Horizon 7 Cloud Connector instance that you specified in step 4.

Step 2 of the setup wizard appears.
10 In this wizard step, provide details about the pod.

These details are used in the Horizon Cloud Administration Console. As an example, the specified name, location, and description are visible in the Administration Console so that you can identify this pod from your other pods that are connected to the control plane.

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

11 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.

When the system determines the pod is successfully connected to the Horizon Cloud control plane, a congratulations message appears.

12 On the final wizard page, click Configure Virtual Center and specify vCenter Server settings.

This configuration enables support for Horizon Cloud to run maintenance actions on the virtual appliance, as needed for successful service operations.

a Enter the FQDN of the vCenter Server, and click Get Certificate. After the certificate information appears, select I have verified the above certificate and would like to continue.

b Under Horizon Cloud Connector vCenter Server Credentials, enter the user name and password for vCenter Server.

c Under Additional Static IP Details for Cloud Connector upgrade, enter the static IP, gateway, subnet mask, and DNS information that you gathered earlier for the Horizon 7 Cloud Connector virtual appliance.

d Click Save.
In addition to **Configure Virtual Center**, several actions are displayed on the final Horizon 7 Cloud Connector Setup wizard page. If you want to perform these actions in the future, you can point your browser to the Horizon 7 Cloud Connector appliance’s IP address like you did in Step 4 and access these options from the wizard page. If you created a forward and reverse lookup record in your DNS server that maps an FQDN to the IP address, you can use that FQDN to access the wizard page.

- If you want to reconfigure the Connection Server details for this same pod, click **Reconfigure** and follow the steps to complete the wizard.
- If you want to remove the connection between this Connection Server instance and the control plane, click **Unplug**.

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

**What to do next**

Log in to the Horizon Cloud Administration Console and view the details about the newly connected pod. For details, see the *Horizon Cloud Administration Guide*.

**Static IP and Proxy Configuration for the Horizon 7 Cloud Connector Virtual Appliance**

Information about the static IP and proxy settings for the Horizon 7 Cloud Connector virtual appliance is saved in certain container files.

**Static IP Storage**

The static IP address of the Horizon 7 Cloud Connector virtual appliance is saved in the `--/opt/container-data/cc-settings/ip.conf` file and is shared with containers running inside the appliance.

For example, the static IP information might appear in the file as follows.

```
cc.address=10.117.163.20
```

**Proxy Storage**

The proxy settings of the Horizon 7 Cloud Connector virtual appliance are stored in the `--/opt/container-data/cc-settings/proxy.conf` and are shared with containers running inside the appliance.

For example, the proxy information might appear in the file as follows.

```
proxyHost=null
proxyPort=0
proxySsl=false
proxyUsername=null
```
Update the Static IP for the Horizon 7 Cloud Connector Virtual Appliance

If you update the static IP address for the Horizon 7 Cloud Connector virtual appliance, you must also perform some additional steps. You must manually reconfigure the container settings file and then send the new static IP information to all desktops associated with the paired Horizon 7 pod.

Prerequisites

Configure a new static IP address for the Horizon 7 Cloud Connector virtual appliance, and join the appliance to Active Directory.

Procedure

1. Edit the `cc.address` line in the `/opt/container-data/cc-settings/ip.conf` file, as shown in the following example.

   ```
   cc.address=[new static IP address]
   ```

2. To send the new static IP address to all running Horizon 7 desktops, run the following command.

   ```
   docker exec csms /bin/bash -c "cd /usr/local/csms; ./scripts/address_changed.sh"
   ```

Configure a CA-Signed Certificate for the Horizon 7 Cloud Connector Virtual Appliance

For enhanced security, you can configure a custom CA-signed certificate for the Horizon 7 Cloud Connector virtual appliance.

Prerequisites

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

Procedure

1. Open an SSH session to the Horizon 7 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.
   - For version 1.4 or later of Horizon 7 Cloud Connector, use the following command:

     ```
     cp /opt/container-data/certs/hze-nginx/server.crt /opt/container-data/certs/hze-nginx/server.crt.orig
     ```
For version 1.3 or earlier of Horizon 7 Cloud Connector, use the following command:

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

5 Back up the existing key.

- For version 1.4 or later of Horizon 7 Cloud Connector, use the following command:

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

- For version 1.3 or earlier of Horizon 7 Cloud Connector, use the following command:

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

6 Copy the existing nginx conf file.

- For version 1.4 or later of Horizon 7 Cloud Connector, use the following command:

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

- For version 1.3 or earlier of Horizon 7 Cloud Connector, use the following command:

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

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

- For version 1.4 or later of Horizon 7 Cloud Connector, use the following command:

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

- For version 1.3 or earlier of Horizon 7 Cloud Connector, use the following command:

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

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

- For version 1.4 or later of Horizon 7 Cloud Connector, use the following command:

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

- For version 1.3 or earlier of Horizon 7 Cloud Connector, 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.

- For version 1.4 or later of Horizon 7 Cloud Connector, 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
```
For version 1.3 or earlier of Horizon 7 Cloud Connector, use the following commands:

```bash
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 block in the nginx configuration file.

- For version 1.4 or later of Horizon 7 Cloud Connector, the `nginx` configuration file is located at `/opt/container-data/conf/hze-nginx/nginx.conf`.
- For version 1.3 or earlier of Horizon 7 Cloud Connector, the `nginx` configuration file is located at `/etc/nginx/nginx.conf`.

11 Verify and restart `nginx`.

- For version 1.4 or later of Horizon 7 Cloud Connector, use the following commands:
  ```bash
docker exec -i hze-nginx sudo nginx -t
systemctl restart hze-nginx
```
- For version 1.3 or earlier of Horizon 7 Cloud Connector, use the following commands:
  ```bash
nginx -t
systemctl restart nginx
```

12 For version 1.4 or later of Horizon 7 Cloud Connector, update the SSL thumbprints in the welcome screen.

Use the following commands:

```bash
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 7 Cloud Connector user interface URL in a Web browser.

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

- For version 1.4 or later of Horizon 7 Cloud Connector, use the following commands:
  ```bash
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
```
- For version 1.3 or earlier of Horizon 7 Cloud Connector, use the following commands:
  ```bash
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
```

**Horizon 7 Cloud Connector Known Considerations**

Keep these considerations in mind when you are using Horizon 7 Cloud Connector.

- Use of IPv6 with the Horizon 7 Cloud Connector virtual appliance is not supported.

- Proxy SSL configuration is not supported during the deployment of the Horizon 7 Cloud Connector virtual appliance.

- Information about the static IP and proxy settings for the Horizon 7 Cloud Connector virtual appliance is saved in certain container files. See Static IP and Proxy Configuration for the Horizon 7 Cloud Connector Virtual Appliance.

- If you update the static IP address for the Horizon 7 Cloud Connector virtual appliance, you must also perform some additional steps to ensure that the new address is shared with the components that depend upon the virtual appliance. See Update the Static IP for the Horizon 7 Cloud Connector Virtual Appliance.

- Before deleting the Horizon 7 Cloud Connector virtual appliance from your vCenter environment, point your browser to the Horizon 7 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 7 Cloud Connector paired with the Horizon 7 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 7 Cloud Connector and Horizon Cloud uses outbound Internet port 443.

- You set the password for the root user of the Horizon 7 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 7 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 7 Cloud Connector interface and use the Reconfigure workflow to perform the certificate validation steps again with the new self-signed certificate. When you log in to the Horizon 7 Cloud Connector interface, you can click Reconfigure and complete the wizard steps to verify communication using the new self-signed certificate from the Connection Server.
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 service. Use the following command:

```bash
systemctl restart hze-core
```

**Set a Password Expiry Policy for the Horizon 7 Cloud Connector Root User**

You can set the password for the root user of the Horizon 7 Cloud Connector virtual appliance during deployment. By default, this password does not expire. However, based on the user's security policy, you might need to periodically update the root password by setting an expiry policy for the root user.

**Note** You must enter all commands as the root user after you log in to the Horizon 7 Cloud Connector virtual appliance. If the user sets a custom password expiry policy, it is your responsibility as an administrator to periodically log in and update the password before it expires. The Horizon 7 Cloud Connector virtual appliance does not notify administrators about password expiry.

**Procedure**

1. To set an expiry policy for the password for the root user, enter the following command:

   ```bash
   chage -M <Max days before password change> -W <Number of days of warning before password expires> root
   ```

   For example, if you want the password to expire after 365 days from the date of password change with a 30-day warning period before the password expires, enter the following command:

   ```bash
   chage -M 365 -W 30 root
   ```

2. To list the current password expiry policy of the root user, enter the following command:

   ```bash
   chage -l root
   ```

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

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

The pod deployed by Horizon Cloud into Microsoft Azure has a physical regional location in a Microsoft Azure cloud. In the pod deployment wizard, you select where to place the pod, according to the regions available for your particular Microsoft Azure subscription. You also select an existing virtual network (VNet) that the pod will use in your selected region.

**Note**  You preconfigure your Microsoft Azure environment with that VNet, and you can either create the subnets required by the pod in advance 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 pod 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 [Optionally Create the 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 Horizon Cloud Administrator Console. The pods you deploy after the first one can reuse the same VNet as your first pod or use different VNets. Also, each pod can be in a different Microsoft Azure region, using a VNet in each region.

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

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

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

- VMs for the pod's manager instance (multiple VMs for a pod that is enabled for high availability)
- VMs for the Unified Access Gateway instances and their load balancers
- VMs for the master RDS-enabled server images
- VMs for the master VDI desktop images
- VMs for the assignable (published) images that are made from the master images
- VMs for the RDSH farms that provide the RDSH desktops and remote applications
- VMs for the VDI desktops
- Additional assets that the VMs and the environment require for supported operations, such as network interfaces, IP addresses, disks, key vaults, Microsoft Azure Database for PostgreSQL server resource, and various items along those lines. The pod deployment process can create the required virtual subnets also, using the values you specify in the deployment wizard.

The following diagram illustrates a deployed pod that is enabled for high availability and has both the external and internal types of gateway configurations. In this diagram, RG means resource group. The Unified Access Gateway instances in the external gateway configuration have NICs on the demilitarized (DMZ) network. When your pod has the external gateway configuration, your end users located in the Internet, outside your corporate network, can access their pod-provisioned virtual desktops and applications through that configuration. When your pod has the internal gateway configuration, your end users located in your intranet, inside your corporate network, can make trusted connections to their pod-provisioned virtual desktops and applications through that gateway. The pod deployment wizard provides the option to deploy the pod with both configurations up front. Alternatively, you can deploy the pod with only one gateway configuration or with none at all, and edit the deployed pod to add the non-chosen gateway configuration later.

You can also choose not to enable the high availability option in the deployment wizard, and then edit the deployed pod later to enable high availability on it. Starting in this release, a new pod is always deployed with a Microsoft Azure Database for PostgreSQL server resource and a pod load balancer, even when you do not enable the high availability option in the wizard. Having those resources available allows for enabling high availability on an already deployed pod. The second pod manager VM is only deployed when high availability is enabled on the pod. For more information, see the High Availability and Your Horizon Cloud Pod in Microsoft Azure documentation topic in the Administration Guide.
Figure 2-1. Illustration of the Horizon Cloud Pod Architecture for a Pod with High Availability Enabled and Configured with Both External and Internal Gateway Configurations, and a Public IP Enabled for the External Gateway’s Load Balancer.
Microsoft Azure Terminology and References

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

**Note** All capitalization and spelling in the citations below follow the same capitalization and spelling found in the linked-to articles in the Microsoft Azure documentation itself.

<table>
<thead>
<tr>
<th>Useful Microsoft Azure References</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Microsoft Azure glossary: A dictionary of cloud terminology on the Azure platform</td>
<td>Use this glossary to learn the meaning of terms as used in the Microsoft Azure cloud context, for terms such as load balancer, region, resource group, subscription, virtual machine, and virtual network (vnet). <strong>Note</strong> The Microsoft Azure glossary does not include the term service principal because the service principal is a resource automatically created in Microsoft Azure when an application registration is created in Microsoft Azure. The reason why you create an application registration in your Microsoft Azure subscription is because that is the way you authorize Horizon Cloud as an application to use your Microsoft Azure capacity. The application registration and its companion service principal enable the Horizon Cloud cloud service acting as an application to access resources in your Microsoft Azure subscription. Use the next reference below to learn about applications and service principals that can access resources in Microsoft Azure.</td>
</tr>
<tr>
<td>Use portal to create an Azure Active Directory application and service principal that can access resources</td>
<td>Use this article to learn about the relationship between an application and a service principal in a Microsoft Azure cloud.</td>
</tr>
<tr>
<td>Azure Resource Manager overview</td>
<td>Use this article to learn about the relationships between resources, resource groups, and the Resource Manager in Microsoft Azure.</td>
</tr>
<tr>
<td>Azure VNet</td>
<td>Use this article to learn about the Azure Virtual Network (VNet) service in Microsoft Azure. See also Azure Virtual Network FAQs.</td>
</tr>
<tr>
<td>Azure VNet Peering</td>
<td>Use this article to learn about virtual network peering in Microsoft Azure.</td>
</tr>
<tr>
<td>Microsoft Azure ExpressRoute Overview</td>
<td>Use this article to learn about Microsoft Azure ExpressRoute and how you can use it to establish connections between your on-premises networks, Microsoft Azure, and your Horizon Cloud pods.</td>
</tr>
<tr>
<td>About VPN Gateway Planning and design for VPN Gateway Create a Site-To-Site connection in the Azure portal</td>
<td>Use these articles to learn about how to configure VPNs in Microsoft Azure.</td>
</tr>
<tr>
<td>What is Azure Load Balancer?</td>
<td>Use this article to learn about the Azure load balancers that are deployed for a pod: the load balancer for the pod manager VMs and the load balancers for the gateway configurations.</td>
</tr>
<tr>
<td>What is Azure Database for PostgreSQL?</td>
<td>Use this article to learn about the Microsoft Azure Database for PostgreSQL service.</td>
</tr>
</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>Considerations on Microsoft Azure with</td>
<td></td>
</tr>
<tr>
<td>VMware Horizon Cloud</td>
<td></td>
</tr>
<tr>
<td>Horizon Cloud Service on Microsoft Azure Security Considerations</td>
<td>Use this article to obtain information about the security details of the environment and the types of data stored.</td>
</tr>
<tr>
<td>Horizon Cloud on Microsoft Azure</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>
<tr>
<td>Desktop and App Scalability (white paper download)</td>
<td></td>
</tr>
</tbody>
</table>

Suggested Workflow for When Your Very First Cloud-Connected Pod is from Deploying into Microsoft Azure

This is a high-level list of the steps for using the wizard in Horizon Cloud to make your very first cloud-connected pod by deploying a pod into your Microsoft Azure capacity. After that first cloud-connected pod is fully deployed and you have completed the steps to register Horizon Cloud with the pod's intended Active Directory domain, you can use all the features provided Horizon Cloud, especially for provisioning VDI desktops, RDSH session-based desktops, or RDSH-based remote applications to your end users from that pod.

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

1. Fulfill the prerequisites, as described in the separate prerequisites checklist document. You can open that document from this PDF link or navigate to it from the Horizon Cloud documentation landing page.

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

3. Deploy the pod. See Deploy a Horizon Cloud Pod into Microsoft Azure.

4. Register your Active Directory domain with the deployed pod, which includes providing the name of a domain-join account. See the Horizon Cloud Administration Guide.

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

   **Important** You must ensure that the domain join account you enter when registering the domain is also in one of the Active Directory groups to which you assign the Horizon Cloud Super Administrators role. The system's domain-join operations depend on the domain join account having the Horizon Cloud Super Administrators role. See the VMware Horizon Cloud Service on Microsoft Azure Administration Guide.
6 Upload SSL certificates to the pod directly, using the pod's summary page in the Administration Console, if you plan to have one or both of the following use cases:

- Use VMware Identity Manager™ with the pod. When you use VMware Identity Manager™ with the pod, you must configure the VMware Identity Manager™ connector to point to the pod itself.
- Clients connecting directly to the pod, not going through a gateway configuration.

See the Horizon Cloud Administration Guide.

**Note** Uploading the SSL certificate to the pod directly is not necessary when your end-user connections go to the Unified Access Gateway instances in the pod through the load balancer connected to those instances.

In general, uploading an SSL certificate to the pod directly is a recommended practice, because that ensures Horizon Clients that might make direct connections to the pod environment can have trusted connections, and is the supported configuration for using VMware Identity Manager™ with the pod. However, connections direct to the pod using HTML Access (Blast) appear as untrusted connections in the end user's browser. The end-user browsers display the typical untrusted certificate error when they make their connections direct to the pod. To have connections using HTML Access (Blast) avoid the displayed untrusted certificate error, you must have those connections use the load balancer and Unified Access Gateway instances from the pod's Unified Access Gateway configuration. If you do not want to expose your fully qualified domain name to the Internet, you can deploy an internal Unified Access Gateway configuration. This internal Unified Access Gateway configuration uses a Microsoft internal load balancer to which end users who are internal to your corporate network can point their connections.

7 Import a master image. On the Imported VMs page, use the Reset Agent Pairing action to pair the new master image with Horizon Cloud. See the Horizon Cloud Administration Guide.

8 Depending on whether your master image is for provisioning VDI desktops or for RDSH-based session desktops and RDSH-based remote applications, perform one or more of the following steps as appropriate. For detailed steps, see the Horizon Cloud Administration Guide.

- In a master image for VDI desktops, install the third-party applications you want your end users to use in their VDI desktops, and configure other applicable customizations, such as setting desktop wallpaper, installing the NVIDIA GPU drivers (for GPU-enabled images), and so on. Also optimize the image for Microsoft Sysprep best practices, if not done as part of the import image process. See the Horizon Cloud Administration Guide.

- In a master RDS-enabled server image for provisioning RDSH-based session desktops and remote applications, install the third-party applications you want to provide to your end users from that RDS image and configure other applicable customizations, such as setting desktop wallpaper, installing the NVIDIA GPU drivers (for GPU-enabled images), and so on. Also optimize the image for Microsoft Sysprep best practices, if not done as part of the import image process.

9 Convert that master image into an assignable image, also known as sealing or publishing the image. See the Horizon Cloud Administration Guide.
10 To provision session-based RDSH desktops and remote applications from a published master server image:
   a Create a desktops RDSH farm to provide session desktops, and then create assignments to entitle end users to use those desktops. See the Horizon Cloud Administration Guide.
   b Create an applications RDSH farm to provide remote applications, add the applications to your application inventory, and then create assignments to entitle end users to use those remote applications. See the Horizon Cloud Administration Guide.

11 To provision VDI desktops from a published master VDI desktop image, create a dedicated or floating VDI desktop assignment. See the Horizon Cloud Administration Guide.

12 When a pod is deployed with a gateway configuration, you must create a CNAME record in your DNS server that maps the fully qualified domain name (FQDN) that you entered in the deployment wizard to the appropriate Azure load balancer resource that is configured in the pod for that gateway.
   ■ For an external gateway enabled with a public IP address, map the FQDN that you entered in the deployment wizard to the gateway's Azure load balancer resource's auto-generated public FQDN. Your DNS server record maps that load balancer's auto-generated public FQDN with the FQDN that your end users will use, and which is used in the uploaded certificate. The following code line demonstrates an example.

     ourApps.ourOrg.example.com   vwm-hcs-podID-uag.region.cloudapp.azure.com

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

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

For details on how to locate the load balancer's FQDN in the Administration Console, see the Horizon Cloud Administration Guide.

13 When a pod is deployed to have RADIUS two-factor authentication for the pod's gateways, you must complete the following tasks:
   ■ If you configured an external gateway with RADIUS settings and that RADIUS server is not reachable within the same VNet as used by the pod, configure that RADIUS server to allow client connections from the IP address of the external gateway's load balancer. In an external gateway configuration, the Unified Access Gateway instances attempt contact with the the RADIUS server using that load balancer address. To allow the connections, ensure the load balancer resource's IP address that is in that external gateway's resource group is specified as a client in your RADIUS server configuration.
   ■ If you configured an internal gateway, or an external gateway and your RADIUS server is reachable within the same VNet as used by the pod, configure the RADIUS server to allow connections from the appropriate NICs that were created in the gateway's resource group in
Microsoft Azure that must communicate with the RADIUS server. Your network administrator determines the RADIUS server’s network visibility to the pod’s Azure Virtual Network and subnets. Your RADIUS server must allow client connections from the IP addresses of those gateway NICs that correspond to the subnet for which your network administrator has given network visibility to the RADIUS server. The gateway’s resource group in Microsoft Azure has four NICs that correspond to that subnet, two that are currently active for the two Unified Access Gateway instances and two that are idle and will become the active ones after the pod goes through an upgrade. To support connectivity between the gateway and the RADIUS server both for ongoing pod operations and after each pod upgrade, ensure the IP addresses of those four NICs are specified as clients in the RADIUS server configuration.

For information on how to obtain those IP addresses, see the Administration Guide.

After the above workflow steps are completed, your end users can launch their entitled desktops and remote applications using your FQDN in the Horizon Client or with HTML Access.

You can find in-depth details on how to accomplish each workflow step in the topics that are linked from each step above or in the companion guide. See the Horizon Cloud Administration Guide.

Preparing to Deploy a Horizon Cloud Pod Into Microsoft Azure

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

1 Fulfill the prerequisites described in the separate prerequisites checklist document, especially:

   ■ Ensure your Microsoft Azure account and subscription encompasses the pod’s required number and sizes of virtual machines. See Microsoft Azure Virtual Machine Requirements for a Horizon Cloud Pod in Microsoft Azure.

   ■ 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.

Important Not all Microsoft Azure regions support GPU-enabled virtual machines. If you want to use the pod for GPU-capable desktops or remote applications, ensure that the Microsoft Azure region you select for the pod provides for those NV-series VM types that you want to use and which are supported in this Horizon Cloud release. See the Microsoft documentation at https://azure.microsoft.com/en-us/regions/services/ for details.
If you want to manually create the subnets for the pod on your VNet in advance of deploying the pod, ensure that the required number of subnets is created on your VNet, that their address spaces meet the pod's requirements, and that they are empty of resources. Optionally Create the 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 Virtual Network's DNS Server.

**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.

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 a service principal and get your Microsoft Azure subscription ID, application ID, application authentication key, and Microsoft Azure AD Directory ID. These resources are used by Horizon Cloud to perform its operations on your Microsoft Azure environment. For detailed steps, see Create the Required Service Principal by Creating an Application Registration.

**Important**  The service principal must have an assigned role in your subscription. The assigned role must allow the actions that Horizon Cloud needs to perform in your Microsoft Azure subscription to successfully deploy the pod and maintain it over time. You must assign to the service principal one of the following roles:

- 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.

- A custom role that you have set up to provide the service principal with Role Operations Required by the Horizon Cloud Pod Deployer in Your Microsoft Azure Subscription that Horizon Cloud needs for pod deployment and maintenance operations.

3. 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 use an external Unified Access Gateway configuration deployed as part of the pod deployment process. 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.

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

4. 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.

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

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

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

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

Pod deployment 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.

**Important** The pod deployment wizard validates that your Microsoft Azure environment has sufficient quota of cores to build the pod. If the wizard determines your subscription does not have sufficient quota, an on-screen message will display and the wizard will not proceed to its next step.

Starting in the September 2019 release, all new pods are deployed with a Microsoft Azure load balancer and Microsoft Azure Database for PostgreSQL server. The Microsoft Azure Database for PostgreSQL server is deployed using the Single Server deployment.

**Note** GPU-enabled VMs are only available in some Microsoft Azure regions. See Microsoft Azure Products by region 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 your subscription’s current quotas in the Microsoft Azure portal, navigate to All services > Subscriptions, click your 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.

Table 2-2. Jump box VM Requirements

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

Note: A jump box VM is newly deployed for creating a pod, for building out an upgrade’s green components when the next version of the pod software is available, for orchestrating the blue/green upgrade process on the pod, and for the process of adding a gateway configuration to an existing pod.
<table>
<thead>
<tr>
<th>VM</th>
<th>Microsoft Azure VM Specification</th>
<th>Quantity</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pod without high availability enabled: Pod management instances</td>
<td>Linux - Standard Dv3 Family: Standard_D4_v3 (4 cores, 16 GB memory). OS disk: Standard HDD 30 GiB</td>
<td>1 per pod during steady-state operations 2 per pod during the end-to-end time for the pod's blue/green upgrade process.</td>
<td>For a pod without high availability enabled, during steady-state operations, one VM exists, is powered on, and runs the pod. When a new pod manifest is made available to you by VMware Operations, and the system begins building out the green components for the pod's blue/green upgrade process, a second instance is created and powered on. As part of the end-to-end upgrade process, you schedule the time at which the system switches to using the green components. After the switch is completed, the pod is using the newly created VM for steady-state operations and the previously used one in the blue component set is stopped and then deleted. Your environment's size must accommodate the two pod manager instances running side-by-side for the end-to-end upgrade time period starting from the time when the system starts building out the pod's green components for the blue/green upgrade process, to when upgrade activities are completed and the pod is switched over to using the new green components. See the Administration Guide for a description of the pod's blue/green upgrade process.</td>
</tr>
<tr>
<td>High-availability-enabled pod: Pod management instances</td>
<td>Linux - Standard Dv3 Family: Standard_D4_v3 (4 cores, 16 GB memory). OS disk: Standard HDD 30 GiB</td>
<td>2 per pod during steady-state operations 4 per pod during the end-to-end pod's blue/green upgrade process.</td>
<td>For a high-availability-enabled pod, during steady-state operations, two VMs exist, are powered on, and run the pod. When a new pod manifest is made available to you by VMware Operations, and the system begins building out the green components for the pod's blue/green upgrade 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 upgrade 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 upgrade time period starting from the time when the system starts building out the pod's green components for the blue/green upgrade process, to when upgrade activities are completed and the pod is switched over to using the new green components. See the Administration Guide for a description of the pod's blue/green upgrade process.</td>
</tr>
<tr>
<td>Unified Access</td>
<td>Linux Standard Av2 Family: Varied based on whether you choose to have an external or internal Unified Access Gateway</td>
<td>Varied based on whether you choose to have an external or internal Unified Access Gateway</td>
<td>Unified Access Gateway is an optional feature that is deployed in your pod when you configure the...</td>
</tr>
</tbody>
</table>
Table 2-3. Pod Management and Unified Access Gateway VM Requirements (continued)

<table>
<thead>
<tr>
<th>VM</th>
<th>Microsoft Azure VM Specification</th>
<th>Quantity</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gateway instances</td>
<td>Standard_A4_v2 Standard (4 cores, 8 GB memory)</td>
<td>Gateway configuration on the pod, or both types.</td>
<td>gateway settings in the deployment wizard. If you decide to have Unified Access Gateway for the pod, your environment’s must accommodate these instances running during the pod’s end-to-end blue/green upgrade 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 upgrade process, two additional instances are created and powered on to run the software updates on Unified Access Gateway. After the upgrade 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 upgrade process, two additional instances per configuration are created and powered on to run the software updates on Unified Access Gateway. After the upgrade 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 upgrade time period starting from the time when the system starts building out the pod’s green components for the blue/green upgrade process, to when upgrade activities are completed and the pod is switched over to using the new green components. See the Administration Guide for a description of the pod’s blue/green upgrade process.</td>
</tr>
<tr>
<td>OS disk: Standard HDD 20 GiB</td>
<td>Access Gateway configuration on the pod, or both types.</td>
<td>2 per pod during steady-state operations</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>4 per pod during the pod’s end-to-end blue/green upgrade process.</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>4 per pod during steady-state operations</td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>8 per pod during the pod’s end-to-end blue/green upgrade process.</td>
<td></td>
</tr>
</tbody>
</table>
### Table 2-4. Image VM Requirements

<table>
<thead>
<tr>
<th>VM</th>
<th>Microsoft Azure VM Specification</th>
<th>Quantity</th>
<th>Description</th>
</tr>
</thead>
</table>
| **Master images**<br>Master images are either GPU-enabled or not, depending on your selection when you create them. | For GPU-enabled master images, the system uses:  
- The NV-series NV6 Standard (from the Standard NV Family vCPUs)<br>- OS disk: Standard HDD 127 GiB | Varied, based on your needs | A master image is a Microsoft Windows operating system VM that is configured so that Horizon Cloud can convert it into a published image. An RDS-enabled Windows server operating system VM provides the base used to create the RDSH farms that provide session-based desktops and remote applications to your end users. A Windows client operating system VM provides the base used to create the VDI desktops. Each master image is a combination of Microsoft Windows operating system and whether it is GPU-enabled or not. So if you want your pod to provide:  
- RDSH desktops using Microsoft Windows 2016 Datacenter, with GPU  
- RDSH desktops using Microsoft Windows 2016 Datacenter, no GPU  
Then you need at least 2 master image VMs.  
The process of converting a master image into a published image is sometimes called publishing the image, or also called sealing the image. The resulting published image is sometimes called a sealed image or an assignable image, because it is in a finalized state for use in assignments.  
The system automatically powers off the master image when it is published (when you perform the **Convert to Image** action on the master image in the Administration Console). When you update a published image, the system powers the VM on again.  
**Note** When you duplicate an image using the Administration Console, the system temporarily powers on the master image's VM to obtain its configuration for the duplicate, and then powers it off again.  
For information about how to create a master image, see the topic Creating Desktop Images for a Horizon Cloud Pod in Microsoft Azure in the Administration Guide. |
| For non-GPU-enabled master images and Microsoft Windows client operating systems, the system uses:  
- The Standard_D4_v3 (from the Standard Dv3 Family vCPUs)<br>- OS disk: Standard HDD 127 GiB | If Microsoft does not provide the Dv3-series in the Microsoft Azure region where you deployed the pod, the system uses the Standard_D3_v2 instead from the Standard Dv2 Family. |
| For non-GPU-enabled master images and Microsoft Windows Server operating systems, the system uses:  
- The Standard_D2_v3 (from the Standard Dv3 Family vCPUs)<br>- OS disk: Standard HDD 127 GiB | If Microsoft does not provide the Dv3-series in the Microsoft Azure region where you deployed the pod, the system uses the Standard_D2_v2 instead from the Standard Dv2 Family. |
<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>Starting in this release, 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. In this release, the OS disk size of the farm's server instances is the same as the OS disk size of the master image on which the farm is based (Standard HDD 127 GiB). For specific details about the Windows VM sizes that are generally available in the standard Microsoft Azure regions, see the Microsoft documentation at <a href="https://docs.microsoft.com/en-us/azure/virtual-machines/windows/sizes">https://docs.microsoft.com/en-us/azure/virtual-machines/windows/sizes</a>. Note For production environments, ensure the VM types you use for your farms have a minimum of two (2) CPUs. Meeting this criteria avoids unexpected end-user connection issues. This criteria is a result of the Horizon Agent recommendations to have a minimum of 2 CPUs to install or upgrade Horizon Agent from version 7.x or later. This Horizon Agent criteria is stated in the Horizon 7 documentation topic Install Horizon Agent on a Virtual Machine from version 7.7 onwards.</td>
<td>Varied, based on your needs and how you have customized the VM sizes in your Horizon Cloud environment.</td>
<td>RDSH farm VMs are the server 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. The power state of these VMs varies, depending on the farm configuration settings and the end-user demand. <strong>Note</strong> In this release, you cannot deliver both session-based desktops and remote applications from the same farm.</td>
</tr>
</tbody>
</table>
Table 2-6. 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>Starting in this release, 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>VDI desktop VMs are the instances that provide VDI desktops to your end users. 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_B2s and Standard_B1s.

In this release, the OS disk size of the VDI desktop instances is the same as the OS disk size of the master image on which the assignment is based (Standard HDD 127 GiB).

For specific details about those Windows VM sizes, see the Microsoft documentation at [https://docs.microsoft.com/en-us/azure/virtual-machines/windows/sizes](https://docs.microsoft.com/en-us/azure/virtual-machines/windows/sizes).

**Note** For production environments, ensure the VM types you use for your VDI desktop assignments have a minimum of two (2) CPUs. Meeting this criteria avoids unexpected end-user connection issues. This criteria is a result of the Horizon Agent recommendations to have a minimum of 2 CPUs to install or upgrade Horizon Agent from version 7.x or later. This Horizon Agent criteria is stated in the Horizon 7 documentation topic [Install Horizon Agent on a Virtual Machine from version 7.7 onwards](https://docs.vmware.com/en-us/horizon/7.7/topic/com.vmware.horizon.doc_7.7/installing_ha_and_ssa.html).
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 server VMs you can deploy in a single subscription and on the total number of concurrent connected sessions you can have per pod. Over time, this topic will be updated to list more of the known limits.

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

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

The 2,000 number per subscription includes VDI desktop VMs and farm server 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 servers. If you have more than one pod in your subscription, the number of VDI desktops and farms servers across all of the pods must not total more than 2,000.

### Maximum of 2,000 sessions per pod

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

Configure the Required Virtual Network in Microsoft Azure

Your Microsoft Azure environment must have an existing virtual network before you can deploy the Horizon Cloud pod into the environment. If you do not already have a virtual network (VNet) in the region into which you are deploying, you must create the virtual network.

In the pod deployment wizard's screens, you will select the VNet and either:

- Specify the address spaces for subnets that the pod deployer will create in the VNet.
- Specify the subnets you created in advance for use by this pod.

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

If you do not want to create the subnets in advance, the pod deployer will automatically create the following subnets on your VNet:

- Management subnet, for IP addresses used by the VMs involved in management activities of the pod itself
Desktop subnet, for IP addresses used for the RDSH server VMs and VDI desktop VMs on that subnet. When the internal Unified Access Gateway configuration is specified in the deployment wizard, the Unified Access Gateway VMs also use IP addresses from this subnet.

**Important**  The VMs for your VDI desktops, the RDS images, and every server in the pod’s RDS farms consume these IP addresses. Because this desktop 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 will 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.

DMZ subnet, for IP addresses used by the optional external Unified Access Gateway configuration. When you have the deployer automatically create the subnets, the deployer always creates the new subnets in the VNet. In terms of the VNet's address space, the deployer handles the subnet address spaces you enter into the wizard as follows:

- 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.
- If the address spaces specified in the wizard are already contained within the VNet's existing address space, the deployer simply creates the new subnets in the VNet using the specified address spaces.

**Important**  If your existing VNet is peered, the deployer cannot automatically update its address space. If the VNet is peered, the best practice is to create the subnets in advance as described in Optionally Create the 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.

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

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

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

**Procedure**

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

   The **Create virtual network** screen appears.
2 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 VNet.</td>
</tr>
<tr>
<td>Address space</td>
<td>Specify the VNet's address space.</td>
</tr>
<tr>
<td>Subscription</td>
<td>Select the same subscription that you are planning to use when you deploy the pod.</td>
</tr>
<tr>
<td>Resource Group</td>
<td>You can either choose an existing resource group or have a new one created when the virtual network is created.</td>
</tr>
<tr>
<td>Location</td>
<td>Select the same region into which you are planning to deploy the pod.</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 Optionally Create the 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.

Retain the default values for the optional settings.

3 Click **Create**.

The virtual network (VNet) is created in your Microsoft Azure account.
What to do next

If you want to manually create 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 Optionally Create the 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 Virtual Network's DNS Server.

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

**Important** The pod's temporary jump box VM and pod's manager VM require outbound Internet access on your Microsoft Azure VNet. If you require proxy-based outbound Internet access, you will need to specify the proxy server information as you complete the fields in the pod deployment wizard.

Optionally Create the Pod's Required Subnets on your VNet in Microsoft Azure

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, both for pods newly deployed in this release and pods upgraded to this release, 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 upgrade 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 desktop 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 has one or more Unified Access Gateway configurations, the Unified Access Gateway VMs also use IP addresses from this subnet.

**Important** The VMs for your VDI desktops, the RDS images, and every server in the pod's RDS farms consume these IP addresses. Because this desktop 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 will 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.

If you are going to have an external Unified Access Gateway configuration, 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 when the external Unified Access Gateway configuration is specified in the deployment wizard. 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.

**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 your VNet that you are using for your pod deployment.
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 Click **OK**.

The subnet is added to the VNet.

6 Repeat steps 3 through 5 to add the remaining required subnets.

The VNet has the required subnets.

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

**What to do next**

For the management subnets you created, enable the Microsoft.Sql service as a service endpoint. See **When Using Existing Subnets for a Horizon Cloud Pod in Microsoft Azure**.

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

Starting with the September 2019 release, both for pods newly deployed in this release and pods upgraded to this release, 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 upgrade 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.

This release introduces 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 upgrade, 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 upgrade 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 upgrade 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.

![Screenshot of enabling Microsoft.Sql as a service endpoint on a subnet](image)

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 Azure**, on the management subnet, you must configure your network rules for the management subnet to allow communication from the pod to the Microsoft Azure Database for PostgreSQL service. You must ensure your management subnets meets this requirement before you deploy a new pod or upgrade 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 Virtual Network’s DNS Server**

The virtual network that you use for Horizon Cloud pod 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.

In a Microsoft Azure subscription, internal network connectivity is not set up by default. For production environments, you 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 virtual network that you plan to use for your pod. See **Configure the Required Virtual Network in Microsoft Azure**.

Ensure that the DNS server you specify in these steps can reach and resolve the specific external names required for a successful pod deployment. For details, see **DNS Requirements for a Horizon Cloud Pod in Microsoft Azure**.
Procedure

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

2. Display the virtual network’s DNS server settings by clicking DNS servers.

3. Using the Custom option, add the address of the DNS server you want to use for name resolution and click Save.

What to do next

Ensure the pod deployer’s access requirements for DNS, ports, and protocols are met. 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 Level.

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 white paper Networking and Active Directory Considerations on Microsoft Azure with VMware Horizon Cloud white paper.

**Important** Your Horizon Cloud environment can consist of pods in Microsoft Azure and Horizon 7 pods on-premises and in VMware Cloud on AWS. As a result, all of those cloud-connected pods must have line of sight to the same set of Active Directory domains. If your environment already has cloud-connected Horizon 7 pods and you are deploying your first pod into Microsoft Azure, you must ensure that pod will be able to have line of sight to the Active Directory domains that are already registered with your Horizon Cloud environment. See the Active Directory-related topics in the Horizon Cloud Administration Guide for more details.

**DNS Requirements for a Horizon Cloud Pod in Microsoft Azure**

For the pod deployment process to deploy your pod successfully into Microsoft Azure, you must configure your firewalls to allow Horizon Cloud to access the Domain Name Service (DNS) addresses it needs. In addition, your DNS must resolve specific names as described in this topic.

**Important** The pod deployment process uses a jump box VM. This jump box VM has ports and protocol requirements for the pod deployment process. See Ports and Protocols Required by the Jump Box During Pod Deployments and Pod Updates.

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

- For a pod created after the September 2019 release or upgraded to that release’s manifest version or later, see Ports and Protocols Requirements for a Horizon Cloud Pod at the September 2019 Release’s Manifest Level. Such pods have manifest versions of 1593 or later.
- For a pod created before the September 2019 release and not yet upgraded to that release’s manifest version, see Ports and Protocols Requirements for a Horizon Cloud Pod Deployed Prior to the September 2019 Release. Such pods have manifest versions of 1493.1 or earlier.

**DNS Requirements for the Pod Deployment Process, Pod Upgrades, and Ongoing Operations**

You must ensure the following DNS names are resolvable and reachable from the pod’s management and tenant subnets using the specific ports and protocols as listed in the following table. Horizon Cloud uses specific outbound ports to securely download the pod software into your Microsoft Azure environment and so that the pod can connect back to the Horizon Cloud control plane. You must configure your network firewall such that Horizon Cloud has the ability to contact the DNS addresses on the ports that it requires. Otherwise, the pod deployment process will fail.
<table>
<thead>
<tr>
<th>Source Pod Subnet</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, depending on which Horizon Cloud control plane is specified in your Horizon Cloud account: cloud.horizon.vmware.com cloud-eu-central-1.horizon.vmware.com cloud-ap-southeast-2.horizon.vmware.com</td>
<td>443</td>
<td>TCP</td>
<td>Horizon Cloud control plane. cloud.horizon.vmware.com is in the United States cloud-eu-central-1.horizon.vmware.com is in Europe cloud-ap-southeast-2.horizon.vmware.com is in Australia</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 agent-related software used in the system's image-related operations.</td>
</tr>
<tr>
<td>Management</td>
<td>d1mes20qfad06k.cloudfront.net</td>
<td>443</td>
<td>TCP</td>
<td>Horizon Cloud content delivery server. On the management subnet, this site is used for downloading the VHDs (virtual hard disks) for the pod's manager and Unified Access Gateway VMs.</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's Linux-based VMs for Ubuntu operating system updates.</td>
</tr>
<tr>
<td>Source Pod Subnet</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>api.snapcraft.io</td>
<td>443</td>
<td>TCP</td>
<td>Ubuntu software package server. Used by the pod's Linux-based VMs for Ubuntu operating system updates.</td>
</tr>
<tr>
<td>Management</td>
<td>archive.ubuntu.com</td>
<td>80</td>
<td>TCP</td>
<td>Ubuntu software package server. Used by the pod's Linux-based VMs for Ubuntu operating system updates.</td>
</tr>
<tr>
<td>Management</td>
<td>changelogs.ubuntu.com</td>
<td>80</td>
<td>TCP</td>
<td>Ubuntu software package server. Used by the pod's Linux-based VMs for tracking Ubuntu operating system updates.</td>
</tr>
<tr>
<td>Management</td>
<td>security.ubuntu.com</td>
<td>80</td>
<td>TCP</td>
<td>Ubuntu software package server. Used by the pod's Linux-based VMs for security-related Ubuntu operating system updates.</td>
</tr>
</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 2-7. Pod Deployment and Operations DNS Requirements (continued)

<table>
<thead>
<tr>
<th>Source Pod Subnet</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. |
| Management        | One of the following, depending on which Microsoft Azure cloud you have deployed your pod into:  
- Microsoft Azure (global): *.blob.core.windows.net  
- Microsoft Azure Germany: *.blob.core.cloudapi.de  
- Microsoft Azure China: *.blob.core.chinacloudapi.cn  
- Microsoft Azure US Government: *.blob.core.usgovcloudapi.net | 443  | TCP   | Used for the pod’s programmatic access to the Azure Blob Storage. Azure Blob Storage is a service for storing large amounts of unstructured object data, such as text or binary data. |
<table>
<thead>
<tr>
<th>Source Pod Subnet</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 have deployed your pod into:  
|                   | Microsoft Azure (global): *.vault.azure.net  
|                   | Microsoft Azure Germany:  
|                   | *.vault.microsoftazure.de  
|                   | Microsoft Azure China: *.vault.azure.cn  
|                   | Microsoft Azure US Government:  
|                   | *.vault.usgovcloudapi.net | 443 | TCP | Used for the pod's ability to programatically work with the Azure Key Vault cloud service. Azure Key Vault is a cloud service that provides a secure store for secrets. |
| Management        | If your firewall or network security group (NSG) supports the use of service tags, one of the following:  
|                   | Global Azure SQL service tag: Sql  
|                   | Region-specific SQL service tag for the Azure region where the pod is deployed:  
|                   | Sql.region, such as Sql.WestUS.  
|                   | If your firewall or network security group (NSG) does not support the use of service tags, you can use the hostname of the database. This name follows the pattern:  
|                   | *.postgres.database.azure.com. | 5432 | TCP | Used for pod communication to the Microsoft Azure PostgreSQL database server. Starting with the September 2019 release, pods that are newly deployed after that release date and pods that are upgraded 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 2-7. Pod Deployment and Operations DNS Requirements (continued)

<table>
<thead>
<tr>
<th>Source Pod Subnet</th>
<th>Destination (DNS name)</th>
<th>Port</th>
<th>Protocol</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tenant</td>
<td>d1mes20qfad06k.cloudfront.net</td>
<td>443</td>
<td>TCP</td>
<td>Horizon Cloud content delivery server. On the tenant subnet, this site is used by the system's automated Import Image process for downloading the installer for the agent-related software.</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Tenant</td>
<td>Depending on which regional Horizon Cloud control plane is specified in your Horizon Cloud account:</td>
<td>443</td>
<td>TCP</td>
<td>Cloud Monitoring Service (CMS)</td>
</tr>
<tr>
<td></td>
<td>North America:</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>kinesis.us-east-1.amazonaws.com</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>query-prod-us-east-1.cms.vmware.com</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Europe:</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>kinesis.eu-central-1.amazonaws.com</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>query-prod-eu-central-1.cms.vmware.com</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Australia:</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>kinesis.ap-southeast-2.amazonaws.com</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>query-prod-ap-southeast-2.cms.vmware.com</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

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

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

Note: A pod that is either deployed new in Microsoft Azure starting with the September 2019 release or which is upgraded to the September 2019 release manifest level and has high availability enabled will have two manager VMs. The following paragraphs use the plural word VMs to indicate the jump box VM must communicate with all of the pod's manager VMs, whether the pod has only one or has two.
During those processes, that jump box VM communicates with the pod's manager VMs using SSH to the manager VMs' port 22. As a result, during the pod deployment process and pod update process, the requirement that communication between the jump box VM and the manager VMs' port 22 must be met. The manager VMs' port 22 must be allowed between the jump box VM as a source and the manager VMs as a destination. Because these VMs are assigned IP addresses dynamically, the network rule to allow this communication should use the management subnet CIDR as both the source and destination, with destination port 22, source port any, and protocol TCP.

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

**Ports and Protocols Requirements for a Horizon Cloud Pod at the September 2019 Release’s Manifest Level**

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

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

**Ports and Protocols Required for Ongoing Operations**

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

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

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

All pods at the September 2019 release's manifest version have a pod Microsoft Azure load balancer. The table rows that involve the pod's load balancer apply for all pods at the manifest level of 1600 or later.
<table>
<thead>
<tr>
<th>Source</th>
<th>Target</th>
<th>Port</th>
<th>Protocol</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Manager VM</td>
<td>Pod's other manager VM</td>
<td>410</td>
<td>TCP</td>
<td>For a pod that is enabled with high availability, this traffic is JMS routing between the manager VMs.</td>
</tr>
<tr>
<td>Pod's Microsoft Azure load balancer</td>
<td>Manager VM</td>
<td>808</td>
<td>HTTP</td>
<td>Health checks of the VMs in the load balancer's backend pool. When a pod at this release's manifest version is not enabled with high availability, the load balancer has one manager VM is its backend pool to check.</td>
</tr>
<tr>
<td>Manager VM</td>
<td>Domain controller</td>
<td>389</td>
<td>TCP, UDP</td>
<td>LDAP services. Server that contains a domain controller role in an Active Directory configuration. Registering the pod with an Active Directory is a requirement.</td>
</tr>
<tr>
<td>Manager VM</td>
<td>Global catalog</td>
<td>326</td>
<td>TCP, UDP</td>
<td>LDAP services. Server that contains global catalog role in an Active Directory configuration. Registering the pod with an Active Directory is a requirement.</td>
</tr>
<tr>
<td>Manager VM</td>
<td>Domain controller</td>
<td>88</td>
<td>TCP, UDP</td>
<td>Kerberos services. Server that contains a domain controller role in an Active Directory configuration. Registering the pod with an Active Directory is a requirement.</td>
</tr>
<tr>
<td>Manager VM</td>
<td>DNS server</td>
<td>53</td>
<td>TCP, UDP</td>
<td>DNS services.</td>
</tr>
<tr>
<td>Manager VM</td>
<td>NTP server</td>
<td>123</td>
<td>UDP</td>
<td>NTP services. Server that provides NTP time synchronization.</td>
</tr>
<tr>
<td>Manager VM</td>
<td>True SSO Enrollment Server</td>
<td>321</td>
<td>TCP</td>
<td>True SSO Enrollment Server. Optional if you are not using True SSO Enrollment Server capabilities with your pods.</td>
</tr>
<tr>
<td>Manager VM</td>
<td>VMware Identity Manager™ service</td>
<td>443</td>
<td>HTTPS</td>
<td>Optional if you are not using VMware Identity Manager™ with the pod. Used to create a trust relationship between the pod and the VMware Identity Manager™ service. Ensure that the pod can reach the VMware Identity Manager™ environment you are using, either on-premises or the cloud service, on port 443. If you are using the VMware Identity Manager™ cloud service, see also the list of VMware Identity Manager™ service IP addresses to which the VMware Identity Manager™ Connector and the pod must have access in the VMware Knowledge Base article 2149884.</td>
</tr>
<tr>
<td>Transient Jump box VM</td>
<td>Manager VM</td>
<td>22</td>
<td>TCP</td>
<td>As described above in Ports and Protocols Required by the Jump Box During Pod Deployments and Pod Updates, a transient jump box is used during pod deployment and pod update processes. Even though ongoing processes do not require these ports, during pod deployment and pod update processes, this jump box VM must communicate with the manager VMs using SSH to the manager VMs' port 22. For details about the cases for which the jump box VM needs this communication, see Ports and Protocols Required by the Jump Box During Pod Deployments and Pod Updates.</td>
</tr>
</tbody>
</table>

**Note** A pod that is at manifest version 1600 or later and has the high availability feature enabled on it, will have two manager VMs. The preceding paragraph uses the plural word VMs to indicate the jump box VM must communicate with all of the pod's manager VMs, whether the pod has only one or has two.
Which ports must be opened for traffic from the end users’ connections to reach their pod-provisioned virtual desktops and remote applications depends on the choice you make for how your end users will connect:

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

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

- **When you do not choose either Unified Access Gateway configurations,** you can instead have your end users connecting directly to the pod, such as using a VPN. For this configuration, you upload an SSL certificate to the pod's manager VMs using the pod's summary page in the Administration Console, as described in the *VMware Horizon Cloud Service Administration Guide*. In general, uploading an SSL certificate to the pod directly is a recommended practice, because that ensures Horizon Clients that might make direct connections to the pod environment can have trusted connections, and is the supported configuration for using VMware Identity Manager™ with the pod. However, connections direct to the pod using HTML Access (Blast) appear as untrusted connections in the end user's browser. The end-user browsers display the typical untrusted certificate error when they make their connections direct to the pod. To have connections using HTML Access (Blast) avoid the displayed untrusted certificate error, you must have those connections use the load balancer and Unified Access Gateway instances from the pod's Unified Access Gateway configuration. If you do
not want to expose your fully qualified domain name to the Internet, you can deploy an internal Unified Access Gateway configuration. This internal Unified Access Gateway configuration uses a Microsoft internal load balancer to which end users who are internal to your corporate network can point their connections.

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

Table 2-9. External End User Connections Ports and Protocols when the Pod Configuration has External Unified Access Gateway instances

<table>
<thead>
<tr>
<th>Source</th>
<th>Target</th>
<th>Port</th>
<th>Protocol</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Horizon Client</td>
<td>Microsoft Azure load balancer for these Unified Access Gateway instances</td>
<td>443</td>
<td>TCP</td>
<td>Login authentication traffic. Can also carry client-drive redirection (CDR), multimedia redirection (MMR), USB redirection, and tunneled RDP traffic. SSL (HTTPS access) is enabled by default for client connections. Port 80 (HTTP access) can be used in some cases. See the topic Understanding What URL Content Redirection Is in the VMware Horizon Cloud Service Administration Guide.</td>
</tr>
<tr>
<td>Horizon Client</td>
<td>Microsoft Azure load balancer for these Unified Access Gateway instances</td>
<td>4172</td>
<td>TCP, UDP</td>
<td>PCoIP via PCoIP Secure Gateway on Unified Access Gateway</td>
</tr>
<tr>
<td>Horizon Client</td>
<td>Microsoft Azure load balancer for these Unified Access Gateway instances</td>
<td>443</td>
<td>TCP</td>
<td>Blast Extreme via Blast Secure Gateway on Unified Access Gateway for data traffic.</td>
</tr>
<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>
<tr>
<td>Browser</td>
<td>Microsoft Azure load balancer for these Unified Access Gateway instances</td>
<td>443</td>
<td>TCP</td>
<td>HTML Access</td>
</tr>
</tbody>
</table>
Table 2-10. Internal End User Connections Ports and Protocols when the Pod Configuration has Internal Unified Access Gateway instances

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

Table 2-11. Internal End User Connections Ports and Protocols when using Direct Pod Connections, Such as Over VPN

<table>
<thead>
<tr>
<th>Source</th>
<th>Target</th>
<th>Port</th>
<th>Protocol</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Horizon Client</td>
<td>Pod's Microsoft Azure load balancer</td>
<td>443</td>
<td>TCP</td>
<td>Login authentication traffic. The traffic from the clients reaches the pod's manager VMs through the pod's load balancer.</td>
</tr>
<tr>
<td>Horizon Client</td>
<td>Horizon agent in the desktop or farm server VMs</td>
<td>4172</td>
<td>TCP, UDP</td>
<td>PCoIP</td>
</tr>
</tbody>
</table>

VMware, Inc.
### Table 2-11. Internal End User Connections Ports and Protocols when using Direct Pod Connections, Such as Over VPN (continued)

<table>
<thead>
<tr>
<th>Source</th>
<th>Target</th>
<th>Port</th>
<th>Protocol</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Horizon Client</td>
<td>Horizon agent in the desktop or farm server VMs</td>
<td>22443</td>
<td>TCP, UDP</td>
<td>Blast Extreme</td>
</tr>
<tr>
<td>Horizon Client</td>
<td>Horizon agent in the desktop or farm server VMs</td>
<td>32111</td>
<td>TCP</td>
<td>USB redirection</td>
</tr>
<tr>
<td>Horizon Client</td>
<td>Horizon agent in the desktop or farm server VMs</td>
<td>9427</td>
<td>TCP</td>
<td>Client-drive redirection (CDR) and multimedia redirection (MMR)</td>
</tr>
<tr>
<td>Browser</td>
<td>Horizon agent in the desktop or farm server VMs</td>
<td>443</td>
<td>TCP</td>
<td>HTML Access</td>
</tr>
</tbody>
</table>

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

### Table 2-12. 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>443</td>
<td>TCP</td>
<td>Login authentication traffic. The traffic from the Unified Access Gateway instances reaches the pod’s manager VM through the pod’s load balancer.</td>
</tr>
<tr>
<td>Unified Access Gateway</td>
<td>Horizon agent in the desktop or farm server VMs</td>
<td>4172</td>
<td>TCP, UDP</td>
<td>PCoIP</td>
</tr>
</tbody>
</table>
| Unified Access Gateway | Horizon agent in the desktop or farm server VMs | 22443 | TCP, UDP | 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. |
| Unified Access Gateway | Horizon agent in the desktop or farm server VMs | 9427  | TCP      | Optional for client driver redirection (CDR) and multimedia redirection (MMR) traffic. |
| Unified Access Gateway | Horizon agent in the desktop or farm server VMs | 32111 | TCP      | Optional for USB redirection traffic.                                    |
| Unified Access Gateway | Your RADIUS instance                        | 1812  | UDP      | When using RADIUS two-factor authentication with that Unified Access Gateway configuration. The default value for RADIUS is shown here. |
The following ports must allow traffic from the Horizon agent-related software that is installed in the desktop VMs and farm server VMs to the high-availability pod's manager VMs.

<table>
<thead>
<tr>
<th>Source</th>
<th>Target</th>
<th>Port</th>
<th>Protocol</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Horizon agent in the desktop or farm server VMs</td>
<td>Manager VM</td>
<td>4002</td>
<td>TCP</td>
<td>Java Message Service (JMS) when using enhanced security (the default)</td>
</tr>
<tr>
<td>Horizon agent in the desktop or farm server VMs</td>
<td>Manager VM</td>
<td>4001</td>
<td>TCP</td>
<td>Java Message Service (JMS), legacy</td>
</tr>
<tr>
<td>Horizon agent in the desktop or farm server VMs</td>
<td>Manager VM</td>
<td>3099</td>
<td>TCP</td>
<td>Desktop message server</td>
</tr>
<tr>
<td>FlexEngine agent (the agent for VMware Dynamic Environment Manager) in the desktop or farm server VMs</td>
<td>Those file shares that you set up for use by the FlexEngine agent that runs in the desktop or farm server VMs</td>
<td>445</td>
<td>TCP</td>
<td>FlexEngine agent access to your SMB file shares, if you are using VMware Dynamic Environment Manager capabilities.</td>
</tr>
</tbody>
</table>

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

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

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

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

**Important** In addition to the ports and protocols described here, you must meet DNS requirements. For details, see DNS Requirements for a Horizon Cloud Pod in Microsoft Azure.
Ports and Protocols Required for Ongoing Operations for a Pod of Manifest Version

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

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

**Table 2-13. 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>Manager VM</td>
<td>Domain controller</td>
<td>389</td>
<td>TCP, UDP</td>
<td>LDAP services. Server that contains a domain controller role in an Active Directory configuration. Registering the pod with an Active Directory is a requirement.</td>
</tr>
<tr>
<td>Manager VM</td>
<td>Global catalog</td>
<td>326</td>
<td>TCP</td>
<td>LDAP services. Server that contains global catalog role in an Active Directory configuration. Registering the pod with an Active Directory is a requirement.</td>
</tr>
<tr>
<td>Manager VM</td>
<td>Domain controller</td>
<td>88</td>
<td>TCP, UDP</td>
<td>Kerberos services. Server that contains a domain controller role in an Active Directory configuration. Registering the pod with an Active Directory is a requirement.</td>
</tr>
<tr>
<td>Manager VM</td>
<td>DNS server</td>
<td>53</td>
<td>TCP, UDP</td>
<td>DNS services.</td>
</tr>
<tr>
<td>Manager VM</td>
<td>NTP server</td>
<td>123</td>
<td>UDP</td>
<td>NTP services. Server that provides NTP time synchronization.</td>
</tr>
<tr>
<td>Manager VM</td>
<td>True SSO Enrollment Server</td>
<td>321</td>
<td>TCP</td>
<td>True SSO Enrollment Server. Optional if you are not using True SSO Enrollment Server capabilities with your pods.</td>
</tr>
<tr>
<td>Manager VM</td>
<td>VMware Identity Manager™ service</td>
<td>443</td>
<td>HTTPS</td>
<td>Optional if you are not using VMware Identity Manager™ with the pod. Used to create a trust relationship between the pod and the VMware Identity Manager™ service. Ensure that the pod can reach the VMware Identity Manager™ environment you are using, either on-premises or the cloud service, on port 443. If you are using the VMware Identity Manager™ cloud service, see also the list of VMware Identity Manager™ service IP addresses to which the VMware Identity Manager™ Connector and the pod must have access in the VMware Knowledge Base article 2149884.</td>
</tr>
<tr>
<td>Transient Jump box VM</td>
<td>Manager VM</td>
<td>22</td>
<td>TCP</td>
<td>As described in Ports and Protocols Required by the Jump Box During Pod Deployments and Pod Updates, a transient jump box is used during pod deployment and pod update processes. Even though ongoing processes do not require these ports, during pod deployment and pod update processes, this jump box VM must communicate with the pod’s manager VM using SSH to the manager VM’s port 22. For details about the cases for which the jump box VM needs this communication, see Ports and Protocols Required by the Jump Box During Pod Deployments and Pod Updates.</td>
</tr>
</tbody>
</table>
Which ports must be opened for traffic from the end users’ connections to reach their pod-provisioned virtual desktops and remote applications depends on the choice you make for how your end users will connect:

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

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

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

In general, uploading an SSL certificate to the pod directly is a recommended practice, because that ensures Horizon Clients that might make direct connections to the pod environment can have trusted connections, and is the supported configuration for using VMware Identity Manager™ with the pod. However, connections direct to the pod using HTML Access (Blast) appear as untrusted connections in the end user's browser. The end-user browsers display the typical untrusted certificate error when they make their connections direct to the pod. To have connections using HTML Access (Blast) avoid the displayed untrusted certificate error, you must have those connections use the load balancer and Unified Access Gateway instances from the pod's Unified Access Gateway configuration. If you do
not want to expose your fully qualified domain name to the Internet, you can deploy an internal Unified Access Gateway configuration. This internal Unified Access Gateway configuration uses a Microsoft internal load balancer to which end users who are internal to your corporate network can point their connections.

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

Table 2-14. External End User Connections Ports and Protocols when the Pod Configuration has External Unified Access Gateway instances

<table>
<thead>
<tr>
<th>Source</th>
<th>Target</th>
<th>Port</th>
<th>Protocol</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Horizon Client</td>
<td>Microsoft Azure load balancer for these Unified Access Gateway instances</td>
<td>443</td>
<td>TCP</td>
<td>Login authentication traffic. Can also carry client-drive redirection (CDR), multimedia redirection (MMR), USB redirection, and tunneled RDP traffic. SSL (HTTPS access) is enabled by default for client connections. Port 80 (HTTP access) can be used in some cases. See the topic Understanding What URL Content Redirection Is in the VMware Horizon Cloud Service Administration Guide.</td>
</tr>
<tr>
<td>Horizon Client</td>
<td>Microsoft Azure load balancer for these Unified Access Gateway instances</td>
<td>4172</td>
<td>TCP, UDP</td>
<td>PCoIP via PCoIP Secure Gateway on Unified Access Gateway</td>
</tr>
<tr>
<td>Horizon Client</td>
<td>Microsoft Azure load balancer for these Unified Access Gateway instances</td>
<td>443</td>
<td>TCP</td>
<td>Blast Extreme via Blast Secure Gateway on Unified Access Gateway for data traffic.</td>
</tr>
<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>
<tr>
<td>Browser</td>
<td>Microsoft Azure load balancer for these Unified Access Gateway instances</td>
<td>443</td>
<td>TCP</td>
<td>HTML Access</td>
</tr>
</tbody>
</table>
Table 2-15. Internal End User Connections Ports and Protocols when the Pod Configuration has Internal Unified Access Gateway instances

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

Table 2-16. Internal End User Connections Ports and Protocols when using Direct Pod Connections, Such as Over VPN

<table>
<thead>
<tr>
<th>Source</th>
<th>Target</th>
<th>Port</th>
<th>Protocol</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Horizon Client</td>
<td>Manager VM</td>
<td>443</td>
<td>TCP</td>
<td>Login authentication traffic</td>
</tr>
<tr>
<td>Horizon Client</td>
<td>Horizon agent in the desktop or farm server VMs</td>
<td>4172</td>
<td>TCP, UDP</td>
<td>PCoIP</td>
</tr>
<tr>
<td>Horizon Client</td>
<td>Horizon agent in the desktop or farm server VMs</td>
<td>22443</td>
<td>TCP, UDP</td>
<td>Blast Extreme</td>
</tr>
</tbody>
</table>
Table 2-16. Internal End User Connections Ports and Protocols when using Direct Pod Connections, Such as Over VPN (continued)

<table>
<thead>
<tr>
<th>Source</th>
<th>Target</th>
<th>Port</th>
<th>Protocol</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Horizon Client</td>
<td>Horizon agent in the desktop or farm server VMs</td>
<td>3211</td>
<td>TCP</td>
<td>USB redirection</td>
</tr>
<tr>
<td>Horizon Client</td>
<td>Horizon agent in the desktop or farm server VMs</td>
<td>9427</td>
<td>TCP</td>
<td>Client-drive redirection (CDR) and multimedia redirection (MMR)</td>
</tr>
<tr>
<td>Browser</td>
<td>Horizon agent in the desktop or farm server VMs</td>
<td>443</td>
<td>TCP</td>
<td>HTML Access</td>
</tr>
</tbody>
</table>

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

Table 2-17. Port Requirements for Traffic from the Pod's Unified Access Gateway Instances

<table>
<thead>
<tr>
<th>Source</th>
<th>Target</th>
<th>Port</th>
<th>Protocol</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Unified Access Gateway</td>
<td>Manager VM</td>
<td>443</td>
<td>TCP</td>
<td>Login authentication traffic</td>
</tr>
<tr>
<td>Unified Access Gateway</td>
<td>Horizon agent in the desktop or farm server VMs</td>
<td>4172</td>
<td>TCP, UDP</td>
<td>PCoIP</td>
</tr>
<tr>
<td>Unified Access Gateway</td>
<td>Horizon agent in the desktop or farm server VMs</td>
<td>22443</td>
<td>TCP, UDP</td>
<td>Blast Extreme By default, when using Blast Extreme, client-drive redirection (CDR) traffic and USB traffic is side-channeled in this port. If you prefer instead, the CDR traffic can be separated onto the TCP 9427 port and the USB redirection traffic can be separated onto the TCP 32111 port.</td>
</tr>
<tr>
<td>Unified Access Gateway</td>
<td>Horizon agent in the desktop or farm server VMs</td>
<td>9427</td>
<td>TCP</td>
<td>Optional for client driver redirection (CDR) and multimedia redirection (MMR) traffic.</td>
</tr>
<tr>
<td>Unified Access Gateway</td>
<td>Horizon agent in the desktop or farm server VMs</td>
<td>3211</td>
<td>TCP</td>
<td>Optional for USB redirection traffic.</td>
</tr>
<tr>
<td>Unified Access Gateway</td>
<td>Your RADIUS instance</td>
<td>1812</td>
<td>UDP</td>
<td>When using RADIUS two-factor authentication with that Unified Access Gateway configuration. The default value for RADIUS is shown here.</td>
</tr>
</tbody>
</table>

The following ports must allow traffic from the Horizon agent-related software that is installed in the desktop VMs and farm server VMs.
<table>
<thead>
<tr>
<th>Source</th>
<th>Target</th>
<th>Port</th>
<th>Protocol</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>Horizon agent in the desktop or farm server VMs</td>
<td>Manager VM</td>
<td>4002</td>
<td>TCP</td>
<td>Java Message Service (JMS) when using enhanced security (the default)</td>
</tr>
<tr>
<td>Horizon agent in the desktop or farm server VMs</td>
<td>Manager VM</td>
<td>4001</td>
<td>TCP</td>
<td>Java Message Service (JMS), legacy</td>
</tr>
<tr>
<td>Horizon agent in the desktop or farm server VMs</td>
<td>Manager VM</td>
<td>3099</td>
<td>TCP</td>
<td>Desktop message server</td>
</tr>
<tr>
<td>FlexEngine agent (the agent for VMware Dynamic Environment Manager) in the desktop or farm server VMs</td>
<td>Those file shares that you set up for use by the FlexEngine agent that runs in the desktop or farm server VMs</td>
<td>445</td>
<td>TCP</td>
<td>FlexEngine agent access to your SMB file shares, if you are using VMware Dynamic Environment Manager capabilities.</td>
</tr>
</tbody>
</table>

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

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

**Create the Required Service Principal by Creating an Application Registration**

Horizon Cloud needs a service principal to access and use your Microsoft Azure subscription’s capacity for your Horizon Cloud pods. When you register a Microsoft Azure AD application, the service principal is also created. Also, you must generate an authentication key and assign a role to the service principal at the subscription level.
For up-to-date and in-depth details and screenshots for creating a service principal, see the Microsoft Azure documentation’s documentation topic How to: Use the portal to create an Azure AD application and service principal that can access resources.

**Important** The service principal must have an assigned role in your subscription. The assigned role must allow the actions that Horizon Cloud needs to perform in your Microsoft Azure subscription to successfully deploy the pod and maintain it over time. You must assign to the service principal one of the following roles:

- 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.
- A custom role that you have set up to provide the service principal with Role Operations Required by the Horizon Cloud Pod Deployer in Your Microsoft Azure Subscription that Horizon Cloud needs for pod deployment and maintenance operations.

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

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

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

- Application ID
- Authentication key

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

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

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

Procedure

1. From the Microsoft Azure portal’s left navigation bar, click Azure Active Directory, then click App registrations.

2. Click New application registration.

3. Type a descriptive name and select a supported account type.
In the Redirect URI section, select **Web**, type `http://localhost:8000`, and click **Register**.

**Register an application**

**Name**
The user-facing display name for this application (this can be changed later).

- [ ] Hzn-Cloud-Principal

**Supported account types**
Who can use this application or access this API?
- [ ] Accounts in this organizational directory only (VMware, Inc.)
- [ ] Accounts in any organizational directory
- [ ] Accounts in any organizational directory and personal Microsoft accounts (e.g. Skype, Xbox, Outlook.com)

Help me choose...

**Redirect URI (optional)**
We’ll return the authentication response to this URI after successfully authenticating the user. Providing this now is optional and it can be changed later, but a value is required for most authentication scenarios.

- [ ] Web
- [ ] http://localhost:8000

By proceeding, you agree to the Microsoft Platform Policies [ ]

**Register**

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

The newly created app registration is displayed on screen.
5 Copy the application ID and directory (tenant) ID and save them to a location where you can retrieve them later when you run the deployment wizard.

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

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

**Caution** You can set the expiration duration to **Never** or to a specific timeframe. However, if you set a specific duration, you must remember to refresh the key before it expires and enter the new key into the pod’s subscription information in the Horizon Cloud Administration Console. Otherwise, the associated pod will stop working. Horizon Cloud cannot detect or know what duration you set.

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

<table>
<thead>
<tr>
<th>DESCRIPTION</th>
<th>EXPIRES</th>
<th>VALUE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hzn-cloud-key1</td>
<td>12/31/2299</td>
<td><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.
7 Assign a role to the service principal at the subscription level.

**Caution** The role can either be the Contributor role or a custom role that you define to permit the required management operations, as described in *Role Operations Required by the Horizon Cloud Pod Deployer in Your Microsoft Azure Subscription*. If the service principal's assigned role does not permit the operations that the pod deployer requires, the deployment wizard will block you from completing the wizard.

a Navigate to your subscription's settings screen by clicking **All services** in the Microsoft Azure portal's main navigation bar, clicking **Subscriptions**, and then clicking the name of the subscription that you will use with the pod.

**Note** At this point, from the screen, you can copy the subscription ID which you will later need in the deployment wizard.

b Click **Access control (IAM)** and then click **Add > Add role assignment** to open the **Add role assignment** screen.

c In the **Add role assignment** screen, for **Role**, select either Contributor or the custom role you created to assign to the service principal.
d. Use the **Select** box to search for your service principal by the name you gave it.

The following screenshot illustrates this step where the Contributor role is selected for the service principal.

![Select box](image)

**Role**: Contributor

**Assign access to**: Azure AD user, group, or application

**Select**: Hzn

**Note**: Make sure the Assign access to drop-down list is set to Azure AD user, group, or application.

e. Click your service principal to make it a selected member and then click **Save**.

![Save button](image)
8 Verify that your subscription has the registered resource providers that the pod requires.

a From the Access control (IAM) screen you are on from the previous step, navigate to the subscription's list of resource providers by clicking (Resource providers) in the subscription's menu.

b Verify that the following resource providers have (Registered) status, and if not, register them.

- Microsoft.Compute
- microsoft.insights
- Microsoft.Network
- Microsoft.Storage
At this point, you’ve created and configured the service provider for the pod, and you have the subscription-related values you need in the first step of the pod deployment wizard. The four subscription-related values are:

- Subscription ID
- Azure Active Directory ID
- Application ID
- Application key value

**What to do next**

Verify that you have collected all of the subscription-related information you will enter in the deployment wizard. See Subscription-Related Information for the Deployment Wizard.

**Role Operations Required by the Horizon Cloud Pod Deployer in Your Microsoft Azure Subscription**

Starting with this release, you can assign a custom role to the service principal instead of using the Microsoft Azure built-in Contributor role. The assigned role must allow the actions that Horizon Cloud needs to perform in your subscription to successfully deploy the pod and maintain it over time, such as the ability to create virtual machines. As a result, if you want to use a custom role for the service principal used by Horizon Cloud, at a minimum, you must define that role to permit the set of management operations described here.

For details about custom roles in Microsoft Azure and the steps you take to create a custom role, see the Microsoft Azure documentation topic Custom roles for Azure resources. For details about how a role works, its structure, and the structure of the management operations, see Understand role definitions for Azure resources in the Microsoft Azure documentation. As described in that documentation topic, a role...
definition is a collection of permissions. This role definition is called the role for short. The role lists the
management operations that can be performed, as well as operations that cannot be performed, by the
service principal to which that role is assigned. A management operation is a combination of the resource
and action performed on that resource.

The pod deployer requires the following actions in the custom role definition. The * (wild card character)
grants access to all operations that match the string within the listed resource provider operation. For the
descriptions of the operations, see the Microsoft Azure documentation at the links listed below.

Table 2-18. Microsoft Azure Resource Operations that Must Be Permitted in the Custom Role

<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 2-18. Microsoft Azure Resource Operations that Must Be Permitted in the Custom Role (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>
<tr>
<td>Operation</td>
<td>Description in the Microsoft Azure Documentation</td>
</tr>
<tr>
<td>----------------------------------------</td>
<td>--------------------------------------------------</td>
</tr>
<tr>
<td>Operation</td>
<td>Description in the Microsoft Azure Documentation</td>
</tr>
<tr>
<td>-----------</td>
<td>-----------------------------------------------</td>
</tr>
</tbody>
</table>
Table 2-18. Microsoft Azure Resource Operations that Must Be Permitted in the Custom Role (continued)

<table>
<thead>
<tr>
<th>Operation</th>
<th>Description in the Microsoft Azure Documentation</th>
</tr>
</thead>
</table>

Example

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 operations that the pod deployer requires. For a description of the properties and usage information, see the Custom role properties section in the Microsoft Azure documentation topic Custom roles for Azure resources. The ID is the unique ID of the custom role. When you create the custom role using Azure PowerShell or Azure CLI, this ID is automatically generated when you create a new role. As described in the Tutorial: Create a custom role for Azure resources using Azure CLI, $mysubscriptionId1$ is the ID of your own subscription.

```json
{
    "Name": "Horizon Cloud Pod",
    "Id": "uuid",
    "IsCustom": true,
    "Description": "Minimum set of Horizon Cloud pod required operations",
    "Actions": [
        "Microsoft.Authorization/*/read",
        "Microsoft.Compute/*/read",
        "Microsoft.Compute/availabilitySets/*",
        "Microsoft.Compute/disks/*",
        "Microsoft.Compute/images/*",
        "Microsoft.Compute/locations/*",
        "Microsoft.Compute/virtualMachines/*",
        "Microsoft.Compute/virtualMachineScaleSets/*",
        "Microsoft.Compute/snapshots/*",
        "Microsoft.DBforPostgreSQL/*",
        "Microsoft.KeyVault/*/read",
        "Microsoft.KeyVault/vaults/*",
        "Microsoft.KeyVault/vaults/secrets/*"
    ]
}
```
Subscription-Related Information for the Deployment Wizard

The Horizon Cloud pod deployment wizard requires you to provide the following pieces of information from your Microsoft Azure subscription.

**Important** You must obtain the application key at the moment you generate it in the Microsoft Azure portal. For information, see Create the Required Service Principal by Creating an Application Registration. You can obtain the other pieces of information at any time by logging in to your Microsoft Azure portal using your Microsoft Azure account credentials.

The IDs are UUIDs, in the form 8-4-4-4-12. These IDs and key described in the following table are used in the first step of the pod deployment wizard.

<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, click <a href="#">Subscriptions</a> in the left menu.</td>
<td></td>
</tr>
<tr>
<td>Directory ID</td>
<td>In the Microsoft Azure portal, click <a href="#">Azure Active Directory</a> &gt; Properties (under <em>Manage</em>).</td>
<td></td>
</tr>
<tr>
<td>Required Value</td>
<td>How to Collect</td>
<td>Your Values</td>
</tr>
<tr>
<td>--------------------</td>
<td>---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------</td>
<td>------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Application ID</td>
<td>In the Microsoft Azure portal, click <a href="#">Azure Active Directory</a>, and then click the application registration that you created for Horizon Cloud using the steps in Create the Required Service Principal by Creating an Application Registration.</td>
<td></td>
</tr>
<tr>
<td>Application Key</td>
<td>Obtain the key when you generate it in the Microsoft Azure portal. See Create the Required Service Principal by Creating an Application Registration.</td>
<td></td>
</tr>
</tbody>
</table>

**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.

- 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.

The resulting PEM file meets the requirements of the pod deployment wizard.

**Deploy a Horizon Cloud Pod into Microsoft Azure**

You run the pod deployment wizard to deploy the components that are collectively called a pod. The pod pairs with Horizon Cloud so that you can use your Microsoft Azure capacity with Horizon Cloud.

The pod deployer uses the information you provide in each step of the wizard to determine how to configure the pod. After you provide the requested information in a particular step, proceed to the next step by clicking **Next**.

**Caution** The IP addresses mentioned in these steps are examples. You should use the address ranges that meet your organization's needs. For each step that mentions an IP address range, substitute ones that are applicable for your organization.
Prerequisites

Before you start the pod deployment wizard, verify that you have the required items. The items you need to provide in the wizard vary according to the pod configuration options you want. For the prerequisites, see Prerequisites for Running the Pod Deployment Wizard.

Pod configuration options include:

- Selecting existing subnets that you create in advance or having the pod deployer automatically create the subnets
- Selecting whether to enable high availability for the pod. If you deploy the pod without high availability enabled, you can later edit the pod to enable it.
- Having the deployment process create a VMware Identity Manager™ tenant.
- Deploying with an external or internal Unified Access Gateway configuration, or deploying with both. If you deploy with only one type of gateway configuration, you can later edit the pod to add the other, non-configured type.

<table>
<thead>
<tr>
<th>If you deploy with the Unified Access Gateway configuration as...</th>
<th>You can later edit the pod to add...</th>
</tr>
</thead>
<tbody>
<tr>
<td>External</td>
<td>Internal</td>
</tr>
<tr>
<td>Internal</td>
<td>External</td>
</tr>
<tr>
<td>None</td>
<td>One or the other or both at the same time</td>
</tr>
</tbody>
</table>

- Deploying with the option for RADIUS two-factor authentication configured on the pod's gateway configurations. If you deploy without RADIUS settings configured on the pod's gateway configurations, you can later edit the pod to add the other, non-configured type.
- For an external gateway configuration, you can optionally select to not have a public IP address on the configuration's load balancer.

**Note** You cannot later edit the deployed pod to change the IP address setting for the external gateway's load balancer.

Procedure

1. **Prerequisites for Running the Pod Deployment Wizard**
   Before you run the pod deployment wizard, verify that your environment satisfies these prerequisites. You must have the following items so that you can provide the requested values in the pod deployment wizard and proceed through the wizard.

2. **Start the Pod Deployment Wizard**
   When deploying your very first pod into Microsoft Azure, you start the pod deployment wizard using the Add Cloud Capacity feature on the Horizon Cloud Administration Console's Getting Started page.
3 Specify the Microsoft Azure Subscription Information for the New 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

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. In this step, you can also optionally select to have the deployment process create a VMware Identity Manager™ tenant.

5 Specify the Horizon Cloud Pod’s Gateway Configuration

In this step of the wizard, specify the information required to deploy the pod with a gateway configured. Unified Access Gateway provides the gateway environment for a pod deployed into Microsoft Azure. When deploying the new pod, you can choose to have an external or internal gateway configuration, or both types on the same pod. You can also deploy the pod without any gateway configuration and decide to add one later after the pod is deployed. By default, when this wizard step displays, the external gateway configuration is selected.

6 Validate and Proceed, and then Start the Pod Deployment Process

After you click Validate & Proceed, the system verifies your specified values. If everything validates, the wizard displays a summary of the information for your review. Then you start the deployment process.

Prerequisites for Running the Pod Deployment Wizard

Before you run the pod deployment wizard, verify that your environment satisfies these prerequisites. You must have the following items so that you can provide the requested values in the pod deployment wizard and proceed through the wizard.

Prerequisites for All Deployments

- Verify that all of the preparatory tasks are completed, as described in Preparing to Deploy a Horizon Cloud Pod Into Microsoft Azure.

- Verify that you have the subscription information, as described in Subscription-Related Information for the Deployment Wizard.

- Verify that you have an existing virtual network in your Microsoft Azure subscription, and in the region in which you are deploying the pod, as described in Configure the Required Virtual Network in Microsoft Azure.

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

- Verify that your VNet is configured to point to a DNS that can resolve external addresses. The pod deployer must be able to reach external addresses in the Horizon Cloud control plane to securely download the pod software into your Microsoft Azure environment.
- Verify that the pod deployer's DNS, ports, and protocols requirements are met, as described in DNS Requirements for a Horizon Cloud Pod in Microsoft Azure and Ports and Protocols Requirements for a Horizon Cloud Pod at the September 2019 Release's Manifest Level.

- If you require use of a proxy for outbound Internet access, verify you have the networking information for your proxy configuration and the authentication credentials it requires, if any. The pod deployment process requires outbound Internet access.

- Verify that you have the information for at least one NTP server that you want the pod to use for time synchronization. The NTP server can be a public NTP server or your own NTP server that you set up for this purpose. The NTP server you specify must be reachable from the virtual network you configured. When you plan to use an NTP server using its domain name instead of a numeric IP address, also ensure that the DNS configured for the virtual network can resolve the NTP server's name.

- If you do not want the deployer to automatically create the subnets it needs, verify that the required subnets have been created in advance and exist on the VNet. For the steps to create the required subnets in advance, see Optionally Create the Pod's Required Subnets on your VNet in Microsoft Azure and When Using Existing Subnets for a Horizon Cloud Pod in Microsoft Azure.

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

- If you are going to have the deployer create the required subnets, verify that you know the address ranges you are going to enter into the wizard for the management subnet, desktop subnet, and DMZ subnet. The DMZ subnet is required when you want the external Unified Access Gateway configuration. Also verify that those ranges do not overlap. You enter the address ranges using CIDR notation (classless inter-domain routing notation). The wizard will display an error if the entered subnet ranges overlap. For the management subnet range, a CIDR of at least /27 is required. For the DMZ subnet range, a CIDR of at least /28 is required. If you want to keep the management and DMZ subnet ranges co-located, you can specify the DMZ subnet range similar to the management subnet with an IP specified. For example, if the management subnet is 192.168.8.0/27, a matching DMZ subnet would be 192.168.8.32/27.

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

- If you are going to have the deployer create the required subnets, verify that subnets with those address ranges do not already exist on the VNet. In this scenario, the deployer itself will automatically
create the subnets using the address ranges you provide in the wizard. If the wizard detects subnets with those ranges already exist, the wizard will display an error about overlapping addresses and will not proceed further. If your VNet is peered, also verify that the CIDR address spaces that you plan to enter in the wizard are already contained in the VNet's address space.

Prerequisites When Creating a VMware Identity Manager™ Cloud Tenant During Pod Deployment

The pod deployment wizard has an option to create a VMware Identity Manager™ tenant in the VMware Identity Manager™ Cloud service as part of the pod deployment. With that option selected, the pod deployment process creates and configures a tenant in the VMware Identity Manager™ Cloud service. After some post-tenant-creation configuration steps, you can use that VMware Identity Manager™ tenant with the pods you deploy in Microsoft Azure using the same Horizon Cloud account.

If you plan to use this option in the wizard, verify that you know:

- The name of the VMware Identity Manager™ data center region in which you want the VMware Identity Manager™ tenant created. In the pod deployment wizard, you will select this data center region from a drop-down menu.
- The name you want to use for your VMware Identity Manager™ tenant.
- A user name that you want for the tenant's admin account.
- Email address. The email address you enter in the wizard will be associated with the tenant's admin account. The welcome email is sent to that email address when the system has created the VMware Identity Manager™ tenant.

A best practice is to use the same email that is the one reflected in the My VMware account that is associated with your VMware Horizon Cloud Service on Microsoft Azure customer account record. This best practice provides for the welcome email about the new tenant going to the same email address where the ones from Horizon Cloud are sent. That is, when you log into the Administration Console to deploy your first pod, you log in with a My VMware name in the form of user@example.com as described in Start the Pod Deployment Wizard. Using that same name as the email address for the VMware Identity Manager™ tenant can make the initial experience easier.

Prerequisites When Deploying With a Unified Access Gateway Configuration

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 going to deploy the pod with both the external and internal Unified Access Gateway configuration types and you want to use the same FQDN for both, you must determine how to route the incoming end-user client traffic to the appropriate load balancer. In this scenario, you need to set up the routing so that client traffic from the Internet is routed to the Microsoft Public Load Balancer and client traffic from your intranet is routed to the Microsoft Internal Load Balancer.

**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 Optionally Create the 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.

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 a Two-Factor Authentication Configuration**

If you are planning to use the two-factor authentication capability, or use it with an on-premises two-factor authentication server, verify that you have the following information used in your authentication server's configuration, so that you can provide it in the appropriate fields in the pod deployment wizard. If you have both a primary and secondary server, obtain the information for each of them.

- IP address or DNS name of the authentication server
- The shared secret that is used for encryption and decryption in the authentication server's protocol messages
- Authentication port numbers, typically the 1812 UDP port.
Authentication protocol type. The authentication types include PAP (Password Authentication Protocol), CHAP (Challenge Handshake Authentication Protocol), MSCHAP1, MSCHAP2 (Microsoft Challenge Handshake Authentication Protocol, version 1 and 2).

**Note** Check your RADIUS vendor’s documentation for the authentication protocol that your RADIUS vendor recommends and follow their indicated protocol type. The pod’s capability to support two-factor authentication with RADIUS is provided by the Unified Access Gateway instances, and Unified Access Gateway supports PAP, CHAP, MSCHAP1, and MSCHAP2. PAP is generally less secure than MSCHAP2. PAP is also a simpler protocol than MSCHAP2. As a result, even though most RADIUS vendors are compatible with the simpler PAP protocol, some RADIUS vendors are not as compatible with the more secure MSCHAP2.

**Start the Pod Deployment Wizard**

When deploying your very first pod into Microsoft Azure, you start the pod deployment wizard using the Add Cloud Capacity feature on the Horizon Cloud Administration Console’s Getting Started page.

**Note** Login authentication into the Horizon Cloud Administration Console relies on My VMware account credentials. If the My VMware account system is experiencing a system outage and cannot take authentication requests, you will not be able to log in to the Administration Console during that time period. If you encounter issues logging in to the Administration Console’s first login screen, check the Horizon Cloud System Status page at https://status.horizon.vmware.com to see the latest system status. On that page, you can also subscribe to receive updates.

**Prerequisites**

Verify that you have met the prerequisites described in Prerequisites for Running the Pod Deployment Wizard.

**Procedure**

1. Log in to the Horizon Cloud Administration Console at https://cloud.horizon.vmware.com using your My VMware account’s credentials.

   The account credentials are the primary email address, such as user@example.com, and the password that are set in the account’s profile.
If you have not previously accepted the Horizon Cloud terms of service using those My VMware credentials, a terms of service notification box appears after you click the Login button. Accept the terms of service to continue.

After signing in, the Horizon Cloud Administration Console opens. When you have no existing pods, the Getting Started wizard is displayed by default with the Capacity section expanded and the Add Cloud Capacity row.

2 In the Add Cloud Capacity row, click Add.

A selection window appears where you can select the cloud into which to deploy this pod.
3. Click **Select** for the Microsoft Azure cloud.

   The Add Cloud Capacity wizard opens to its first step.

<table>
<thead>
<tr>
<th>Add Microsoft Azure Capacity</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Subscription</td>
</tr>
<tr>
<td>2. Pod Setup</td>
</tr>
<tr>
<td>3. Gateway Settings</td>
</tr>
<tr>
<td>4. Summary</td>
</tr>
</tbody>
</table>

   Choose the Microsoft Azure subscription you want to apply or add a new one.

   **Microsoft Azure Subscription Details**

   - **Apply Subscription:**
   - **Add New:**
   - **Subscription Name:**
   - **Environment:**
   - **Subscription ID:**
   - **Directory ID:**
   - **Application ID:**
   - **Application Key:**

4. Specify the subscription to use for this pod by following the steps in **Specify the Microsoft Azure Subscription Information for the New Pod**.

**Specify the Microsoft Azure Subscription Information for the New Pod**

In this step of the pod deployment wizard, you provide the Microsoft Azure subscription information that you want to use for this pod.

**Prerequisites**

- Verify that you have met the prerequisites described in **Deploy a Horizon Cloud Pod into Microsoft Azure**.
- For this wizard step, verify that you have the subscription-related information as described in **Subscription-Related Information for the Deployment Wizard**.
- Complete the steps in **Start the Pod Deployment Wizard**.
**Procedure**

1. On the wizard’s first step, specify the subscription to use for this pod by selecting the name of a previously entered subscription or entering new subscription information.

   Add Microsoft Azure Capacity

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Apply Subscription</td>
<td>Select the name of a previously entered subscription or select <strong>Add New</strong> to enter new subscription information.</td>
</tr>
<tr>
<td>Subscription Name</td>
<td>When providing new subscription information, enter a friendly name so you can identify this subscription from other previously entered subscriptions. The name must start with a letter and contain only letters, dashes, and numbers.</td>
</tr>
</tbody>
</table>

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>Environment</td>
<td>Select the cloud environment associated with your subscription, for example:</td>
</tr>
<tr>
<td></td>
<td>- Azure, for the standard global Microsoft Azure cloud</td>
</tr>
<tr>
<td></td>
<td>- Azure - China, for the Microsoft Azure in China cloud</td>
</tr>
<tr>
<td></td>
<td>- Azure - Germany, for the Microsoft Azure Germany cloud</td>
</tr>
<tr>
<td></td>
<td>- Azure - US Government, for the Microsoft Azure US Government cloud</td>
</tr>
<tr>
<td>Subscription ID</td>
<td>Enter your cloud capacity subscription ID (in UUID form). This subscription ID must be valid for the environment you selected. For Microsoft Azure, you can obtain this UUID from your Microsoft Azure portal's Subscriptions area.</td>
</tr>
<tr>
<td>Directory ID</td>
<td>Enter your Microsoft Azure AD Directory ID (in UUID form). For Microsoft Azure, you can obtain this UUID from your Microsoft Azure Active Directory properties in the Microsoft Azure portal.</td>
</tr>
<tr>
<td>Application ID</td>
<td>Enter the application ID (in UUID form) associated with the service principal you created in the Microsoft Azure portal. Creating an application registration and its associated service principal in your Microsoft Azure Active Directory is a prerequisite.</td>
</tr>
<tr>
<td>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>
</tbody>
</table>

**Important** In this screen, you cannot delete previously entered subscription values associated with a particular Subscription Name. Even though this occurrence is rare, you might imagine a situation where you:

a. Set up the subscription-related pieces in Microsoft Azure.

b. Start the Add Cloud Capacity wizard, enter those subscription values in the first step, and progress on to the next wizard step.

c. However, upon reading the networking values requested in the next wizard step, you cancel out of this wizard and open a new browser tab to go into the Microsoft Azure portal and adjust your networking configuration to meet the prerequisites.

d. While in the Microsoft Azure portal, you then also decide to make a new application registration to have a service provider with a different name.

e. You return to the browser that has the Getting Started page and restart the Add Cloud Capacity wizard.

At this point, your previously entered subscription name is still in the Apply Subscription drop-down list. However, if you select that name, all of the fields are prepopulated with the previous values, including the old application ID, and there is not a way to change the values in the screen or edit or delete that subscription name to start over with it. If this happens to you, cancel out of the wizard, restart the wizard, and on this first step, create a brand new subscription name by selecting Add New, enter the current values you want to use, and proceed forward. Later on, when the pod is fully deployed and you can navigate to its summary page in the Administration Console, you will be able to delete the old subscription information at that time.

The following screenshot is an example with this step completed.
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.
- Does the application's service principal for the specified application ID have a role that permits all of the operations that the pod deployment process requires. For a description of the service principal and its role requirements, see the topic Create the Required Service Principal by Creating an Application Registration in the Deployment Guide.

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.
Specify Pod Configuration Information

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. In this step, you can also optionally select to have the deployment process create a VMware Identity Manager™ tenant.

**Caution** The IP addresses mentioned in these steps are examples. You should use the address ranges that meet your organization's needs. For each step that mentions an IP address range, substitute ones that are applicable for your organization.

**Prerequisites**

Verify that you have met the prerequisites described in Prerequisites for Running the Pod Deployment Wizard.

If you are going to have the deployment process automatically create the required subnets, verify that the CIDR address ranges that you plan to specify in the wizard fields for those subnets are not already being used by existing subnets on your VNet in Microsoft Azure.

If you have created subnets in advance for use with this pod, verify those subnets have no resources attached to them and verify that the subnet you created to use for the management subnet has the Microsoft.SQL service configured as a service endpoint for that subnet. The pod deployment wizard will validate that the Microsoft.SQL service is configured as a service endpoint on the management subnet.

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

**Procedure**

1. In this step of the wizard, provide details about the pod and the required networking information.

   The following screenshot is an example of the step when it is initially displayed.
Add Microsoft Azure Capacity

Enter your pod details here to configure and connect it.

Details

Pod Name: 

Location: 
Add New

Microsoft Azure Region: 
Select

Description:

High Availability

Enabled:

Networking

Virtual Network: 

Use Existing Subnet:

Management Subnet (CIDR):

Desktop Subnet (CIDR):

NTP Servers:

Use Proxy:

Identity Management

Identity Manager Tenant:

Option | Description
--- | ---
Pod Name | Enter a friendly name for this pod. This name is used in the Administration Console to identify this pod from your other pods.

Location | 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 Administration 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.

Note You must select a city from the system's autocomplete list.
<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Microsoft Azure Region</strong></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.</td>
</tr>
<tr>
<td><strong>Important</strong></td>
<td>Not all Microsoft Azure regions support GPU-enabled virtual machines. If you want to use the pod for GPU-capable desktops or remote applications, ensure that the Microsoft Azure region you select for the pod provides for those NV-series VM types that you want to use and which are supported in this Horizon Cloud release. See the Microsoft documentation at <a href="https://azure.microsoft.com/en-us/regions/services/">https://azure.microsoft.com/en-us/regions/services/</a> for details.</td>
</tr>
<tr>
<td><strong>Description</strong></td>
<td>Optional: Enter a description for this pod.</td>
</tr>
<tr>
<td><strong>High Availability</strong></td>
<td>Enable this toggle to deploy a pod that is configured with high availability. For details about high availability and the pod, see the Administration Guide. If you disable this toggle, the pod is deployed without high availability.</td>
</tr>
<tr>
<td><strong>Virtual Network</strong></td>
<td>Select a virtual network from the list. Only virtual networks (V Nets) that exist in the region selected in the <strong>Microsoft Azure Region</strong> field are shown here. You must have already created the VNet you want to use in that region in your Microsoft Azure subscription.</td>
</tr>
<tr>
<td><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. <strong>Important</strong> 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 of the pod's required subnets.</td>
</tr>
<tr>
<td><strong>Management Subnet</strong></td>
<td>When <strong>Use Existing Subnet</strong> is enabled, <strong>Management Subnet</strong> 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. <strong>Important</strong> Select a subnet that has the Microsoft.Sql service configured as a service endpoint for that subnet. This service endpoint supports the required communication between the pod manager VMs and the pod's Azure Postgres database over the management subnet. 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>Management Subnet (CIDR)</strong></td>
<td>When <strong>Use Existing Subnet</strong> is disabled, in <strong>Management Subnet (CIDR)</strong>, enter a subnet address range (in CIDR notation) for the deployer to create a subnet to which the pod and Unified Access Gateway instances will get connected, such as 192.168.8.0/27. For the management subnet, a CIDR of at least /27 is required. <strong>Caution</strong> When you do not select the wizard option to use existing subnets, the subnet must not already exist in your Microsoft Azure environment. If it already exists, you will get an error when you try to proceed to the next wizard step.</td>
</tr>
<tr>
<td>Option</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>---------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Desktop Subnet</td>
<td>When Use Existing Subnet is enabled, Desktop 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 desktop tenant subnet.</td>
</tr>
<tr>
<td>(CIDR)</td>
<td><strong>Important</strong> Select an empty subnet, one that has no other resources attached to it. If the subnet is not empty, unexpected results might occur during the deployment process or pod operations.</td>
</tr>
<tr>
<td></td>
<td>When Use Existing Subnet is disabled, in Desktop Subnet (CIDR), enter a subnet address range (in CIDR notation) for the deployer to create a subnet to which all of this pod's VDI desktops and RDSH farm servers for end-user remote desktops and applications will get connected, such as 192.168.12.0/22. For the desktop subnet, a CIDR of at least /27 is required, and a CIDR of /22 is recommended.</td>
</tr>
<tr>
<td></td>
<td><strong>Important</strong> Ensure the range you enter is large enough to allow for accommodating the number of desktops you anticipate you will want this pod to provide. This desktop subnet cannot be extended after the pod is deployed.</td>
</tr>
<tr>
<td></td>
<td><strong>Caution</strong> When you do not select the wizard option to use existing subnets, the subnet must not already exist in your Microsoft Azure environment. If it already exists, you will get an error when you try to proceed to the next wizard step.</td>
</tr>
<tr>
<td>NTP Servers</td>
<td>Enter the list of NTP servers you want to use for time synchronization, separated by commas.</td>
</tr>
<tr>
<td></td>
<td>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.</td>
</tr>
<tr>
<td></td>
<td>Examples of public NTP server domain names are time.windows.com, us.pool.ntp.org, time.google.com.</td>
</tr>
<tr>
<td>Use Proxy</td>
<td>If you require a proxy for outbound Internet connectivity, enable this toggle and complete the associated displayed fields.</td>
</tr>
<tr>
<td></td>
<td>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.</td>
</tr>
<tr>
<td></td>
<td>■ <strong>Proxy</strong> (required): Type the hostname or IP address for your proxy server.</td>
</tr>
<tr>
<td></td>
<td>■ <strong>Port</strong> (required): Type the port number that is specified in your proxy server configuration.</td>
</tr>
<tr>
<td></td>
<td>If your proxy server configuration requires a user name and password for authentication, provide those credentials also.</td>
</tr>
</tbody>
</table>

The following screenshot is an example with this step completed when having the deployment process automatically create the subnets and with **High Availability** enabled. In this example, a proxy was not needed to meet the outbound Internet connectivity requirement.
In the Identity Management section, if you want to have the pod deployment process create a VMware Identity Manager™ cloud tenant, enable the Identity Manager Tenant toggle and complete the associated displayed fields.

**Important** The system can create the VMware Identity Manager™ tenant in this workflow of creating the pod. In this release, you cannot later edit an existing pod to have the system create the VMware Identity Manager™ tenant for you. If you create the pod with the Identity Manager Tenant toggle enabled, and then later want to use a VMware Identity Manager™ tenant with the Horizon Cloud pods in this customer account, you must obtain your tenant by subscribing to the VMware Identity Manager™ cloud-hosted environment.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Data Center Region</td>
<td>Select the VMware Identity Manager™ region for the new tenant.</td>
</tr>
<tr>
<td>Tenant Name</td>
<td>Type a name for the tenant.</td>
</tr>
<tr>
<td>Option</td>
<td>Description</td>
</tr>
<tr>
<td>----------</td>
<td>-------------</td>
</tr>
<tr>
<td>Username</td>
<td>Type a name to use for the admin account of the VMware Identity Manager™ tenant.</td>
</tr>
<tr>
<td>Email</td>
<td>Type the email address to use for the admin account in Username. The welcome email is sent to that email address when the system has created the VMware Identity Manager™ tenant. A best practice is to use the same email that is the one reflected in the My VMware account that is associated with your VMware Horizon Cloud Service on Microsoft Azure customer account record. This best practice provides for the welcome email about the new tenant going to the same email address where the ones from Horizon Cloud are sent. That is, when you log into the Administration Console to deploy your first pod, you log in with a My VMware name in the form of <a href="mailto:user@example.com">user@example.com</a> as described in Start the Pod Deployment Wizard. Using that same name as the email address for the VMware Identity Manager™ tenant can make the initial experience easier.</td>
</tr>
</tbody>
</table>

The following screenshot shows the Identity Management section with the fields filled out.

![Identity Management Screenshot]

**Note** After the pod is deployed, further steps are required for integrating the new VMware Identity Manager™ tenant with your pod. For those subsequent steps, see the Horizon Cloud Administration Guide and its Integrate a Horizon Cloud Pod with a VMware Identity Manager™ Environment topic and subtopics. The pod deployer does not deploy the VMware Identity Manager™ connector appliance that is required for the pod to integrate with the VMware Identity Manager™ service.

3 Proceed to the next step by clicking **Next**.

4 Specify details for the pod to have a Unified Access Gateway configuration by following the steps in Specify the Horizon Cloud Pod's Gateway Configuration. For the ability to have your end users access their desktops and remote applications over the Internet, an external Unified Access Gateway configuration is required.

**Specify the Horizon Cloud Pod's Gateway Configuration**

In this step of the wizard, specify the information required to deploy the pod with a gateway configured. Unified Access Gateway provides the gateway environment for a pod deployed into Microsoft Azure. When deploying the new pod, you can choose to have an external or internal gateway configuration, or both types on the same pod. You can also deploy the pod without any gateway configuration and decide...
to add one later after the pod is deployed. By default, when this wizard step displays, the external gateway configuration is selected.

**External gateway configuration**

The external Unified Access 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.

**Internal gateway configuration**

The internal Unified Access 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.
Prerequisites

Verify that you have met the prerequisites described in Prerequisites for Running the Pod Deployment Wizard.

Important To complete this step, you must have the required fully qualified domain name (FQDN) which your end users will use to access the service and have a signed SSL certificate (in PEM format) based on that FQDN. The certificate must be signed by a trusted CA. A single PEM file must contain the entire certificate chain and the private key: SSL certificate intermediate certificates, root CA certificate, private key. For details, see Convert a Certificate File to the PEM Format Required for Pod Deployment.

Verify that all certificates in the certificate chain have valid time frames. If any certificate in the chain is expired, unexpected failures can occur later in the pod onboarding process.

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

1. If you want the external gateway configuration, complete the fields in the **External UAG** section.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enable External UAG?</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 an Azure load balancer resource and Unified Access Gateway instances to provide this access.</td>
</tr>
<tr>
<td><strong>Note</strong></td>
<td>Leaving the default enabled setting is recommended.</td>
</tr>
<tr>
<td>DMZ Subnet</td>
<td>When <strong>Use Existing Subnet</strong> is enabled in the preceding wizard step, <strong>DMZ Subnet</strong> lists the subnets available on the VNet selected for Virtual Network. Select the existing subnet that you want to use for the pod's DMZ subnet.</td>
</tr>
<tr>
<td><strong>Important</strong></td>
<td>Select an empty subnet, one that has no other resources attached to it. If the subnet is not empty, unexpected results might occur during the deployment process or pod operations.</td>
</tr>
<tr>
<td>DNS Addresses</td>
<td>Optionally enter addresses for additional DNS servers that Unified Access Gateway can use for name resolution, separated by commas. When configuring this external Unified Access Gateway configuration to use two-factor authentication with your on-premises RADIUS server, you would specify the address of a DNS server that can resolve the name of your on-premises RADIUS server. As described in the Prerequisites for All Deployments, a DNS server must be set up internally in your subscription and configured to provide external name resolution. The Unified Access Gateway instances use that DNS server by default. If you specify addresses in this field, the deployed Unified Access Gateway instances use the addresses in addition to the prerequisite DNS server that you configured in your subscription's virtual network.</td>
</tr>
<tr>
<td>Routes</td>
<td>Optionally specify custom routes to additional gateways that you want the deployed Unified Access Gateway instances to use to resolve network routing for the end user access. The specified routes are used to allow Unified Access Gateway to resolve network routing such as to RADIUS servers for two-factor authentication. When configuring this pod to use two-factor authentication with an on-premises RADIUS server, you must enter the correct route the Unified Access Gateway instances can use to reach the RADIUS server. For example, if your on-premises RADIUS server uses 10.10.60.20 as its IP address, you would enter 10.10.60.0/24 and your default route gateway address as a custom route. You obtain your default route gateway address from the Express Route or VPN configuration you are using for this environment. Specify the custom routes as a comma-separated list in the form ipv4-network-address/bits ipv4-gateway-address, for example: 192.168.1.0/24 192.168.0.1, 192.168.2.0/24 192.168.0.2.</td>
</tr>
</tbody>
</table>
Option | Description
--- | ---
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.
Enable Public IP? | Controls whether this gateway's load balancing type is configured as private or public. If switched on, the deployed Azure load balancer resource is configured with a public IP address. If switched off, the Azure load balancer resource is configured with a private IP address.

**Important** 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.

The following screenshot is an example with this step completed.

2 (Optional) In the **External UAG** section, optionally configure two-factor authentication for the external Unified Access Gateway.

Complete the steps in **Specify Two-Factor Authentication Capability for the Pod.**
In the Internal UAG section, if you want the internal Unified Access Gateway configuration, switch on the Enable Internal UAG? toggle and complete the fields that appear.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Enable Internal UAG?</td>
<td>Controls whether the pod has an internal gateway configuration. The internal configuration provides trusted access to desktops and applications for HTML Access (Blast) connections for users located inside of your corporate network. The pod includes an Azure load balancer resource and Unified Access Gateway instances to provide this access. By default, this gateway’s load balancing type is private. The load balancer is configured with a private IP address.</td>
</tr>
<tr>
<td>FQDN</td>
<td>Enter the required fully qualified domain name (FQDN), such as ourOrg.example.com, which your end users will use to access the service. You must own that domain name and have a certificate in PEM format that can validate that FQDN. 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 internal Unified Access Gateway configuration to use two-factor authentication with your on-premises RADIUS server, you would specify the address of a DNS server that can resolve the name of your on-premises RADIUS server. As described in the Prerequisites for All Deployments, a DNS server must be set up internally in your subscription and configured to provide name resolution. The Unified Access Gateway instances use that DNS server by default. If you specify addresses in this field, the deployed Unified Access Gateway instances use the addresses in addition to the prerequisite DNS server that you configured in your subscription’s virtual network.</td>
</tr>
<tr>
<td>Routes</td>
<td>Optionally specify custom routes to additional gateways that you want the deployed Unified Access Gateway instances to use to resolve network routing for the end user access. The specified routes are used to allow Unified Access Gateway to resolve network routing such as to RADIUS servers for two-factor authentication. When configuring this pod to use two-factor authentication with an on-premises RADIUS server, you must enter the correct route the Unified Access Gateway instances can use to reach the RADIUS server. For example, if your on-premises RADIUS server uses 10.10.60.20 as its IP address, you would enter 10.10.60.0/24 and your default route gateway address as a custom route. You obtain your default route gateway address from the Express Route or VPN configuration you are using for this environment. Specify the custom routes as a comma-separated list in the form ipv4-network-address/bits ipv4-gateway-address, for example: 192.168.1.0/24 192.168.0.1, 192.168.2.0/24 192.168.0.2.</td>
</tr>
<tr>
<td>Certificate</td>
<td>Upload the certificate in PEM format that Unified Access Gateway will use to allow clients to trust connections to the Unified Access Gateway instances running in Microsoft Azure. The certificate must be based on the FQDN you entered and be signed by a trusted CA. The PEM file must contain the entire certificate chain and the private key: SSL certificate intermediate certificates, root CA certificate, private key.</td>
</tr>
</tbody>
</table>

(Optional) In the Internal UAG section, optionally configure two-factor authentication for the internal Unified Access Gateway.

Complete the steps in Specify Two-Factor Authentication Capability for the Pod.

When you have provided the required information associated with your selected options, you can click Validate & Proceed to continue to the wizard’s final step. See Validate and Proceed, and then Start the Pod Deployment Process.

Specify Two-Factor Authentication Capability for the Pod
In the pod deployment wizard step for specifying its Unified Access Gateway configurations, you can also specify use of two-factor authentication for your end users’ access to their desktops and applications.
through those gateway configurations. You can specify these two-factor authentication details after providing the Unified Access Gateway configuration details.

**Prerequisites**

Verify that you have met the prerequisites described in Prerequisites for Running the Pod Deployment Wizard.

For the external or internal Unified Access Gateway configuration for which you are entering the two-factor authentication details, verify that you have completed the fields for the Unified Access Gateway configuration in the wizard as described in Specify the Horizon Cloud Pod's Gateway Configuration.

When configuring two-factor authentication to an on-premises authentication server, you also provide information in the following fields so that the Unified Access Gateway instances can resolve routing to that on-premises server.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DNS Addresses</td>
<td>Specify one or more addresses of DNS servers that can resolve the name of your on-premises authentication server.</td>
</tr>
<tr>
<td>Routes</td>
<td>Specify one or more custom routes that allow the pod's Unified Access Gateway instances to resolve network routing to your on-premises authentication server. For example, if you have an on-premises RADIUS server that uses 10.10.60.20 as its IP address, you would use 10.10.60.0/24 and your default route gateway address as a custom route. You obtain your default route gateway address from the Express Route or VPN configuration you are using for this environment. Specify the custom routes as a comma-separated list in the form ipv4-network-address/bits ipv4-gateway-address, for example: 192.168.1.0/24 192.168.0.1, 192.168.2.0/24 192.168.0.2.</td>
</tr>
</tbody>
</table>

Verify that you have the following information used in your authentication server's configuration, so that you can provide it in the appropriate fields in the pod deployment wizard. If you have both a primary and secondary server, obtain the information for each of them.

- IP address or DNS name of the authentication server
- The shared secret that is used for encryption and decryption in the authentication server's protocol messages
- Authentication port numbers, typically the 1812 UDP port.
- Authentication protocol type. The authentication types include PAP (Password Authentication Protocol), CHAP (Challenge Handshake Authentication Protocol), MSCHAP1, MSCHAP2 (Microsoft Challenge Handshake Authentication Protocol, version 1 and 2).

**Note**  Check your RADIUS vendor's documentation for the authentication protocol that your RADIUS vendor recommends and follow their indicated protocol type. The pod's capability to support two-factor authentication with RADIUS is provided by the Unified Access Gateway instances, and Unified Access Gateway supports PAP, CHAP, MSCHAP1, and MSCHAP2. PAP is generally less secure than MSCHAP2. PAP is also a simpler protocol than MSCHAP2. As a result, even though most RADIUS vendors are compatible with the simpler PAP protocol, some RADIUS vendors are not as compatible with the more secure MSCHAP2.
**Procedure**

1. **Switch on the Enable 2 Factor Authentication toggle.**

   When the toggle is enabled, the wizard displays the additional configuration fields. Use the scroll bar to access all of the fields.

   The following screenshot is an example of what is displayed after you switch on the toggle in the **External UAG** section.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>DMZ Subnet (CIDR):</td>
<td>192.168.20.32/28</td>
</tr>
<tr>
<td>DNS Addresses:</td>
<td>192.168.0.15</td>
</tr>
<tr>
<td>Routes:</td>
<td></td>
</tr>
<tr>
<td>Certificate:*</td>
<td>ourOrg.pem</td>
</tr>
<tr>
<td>Enable Public IP?</td>
<td></td>
</tr>
</tbody>
</table>

2. **Select your two-factor authentication method in the drop-down list.**

   In this release, RADIUS authentication is supported.

3. **In the Name field, enter an identifying name for this configuration.**

4. **In the Properties section, specify details related to the end users' interaction with the login screen they will use to authenticate for access.**

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>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 Unified Access Gateway. 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 Enter your <strong>DisplayHint</strong> user name and passcode, 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</strong> for would result in a prompt to the end user that says Enter your Example Company user name and domain password below for user name and passcode.</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 the system will append to the SAML assertion user name that is sent to the broker. For example, if the user name is entered as user1 on the login screen and a name ID suffix of @example.com was specified here, the system sends a SAML assertion user name of <a href="mailto:user1@example.com">user1@example.com</a> to the broker.</td>
</tr>
<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>
<tr>
<td>Maintain Username</td>
<td>Enable this toggle to maintain the user's RADIUS username during authentication to Horizon Cloud. When enabled:</td>
</tr>
<tr>
<td></td>
<td>- The user must have the same username credentials for RADIUS as for their Active Directory authentication to Horizon Cloud.</td>
</tr>
<tr>
<td></td>
<td>- The user cannot change the username in the login screen.</td>
</tr>
<tr>
<td></td>
<td>If this toggle is disabled, the user is able to type a different user name in the login screen.</td>
</tr>
<tr>
<td>Note</td>
<td>For the relationship between enabling Maintain Username and the domain security settings in Horizon Cloud, see the Domain Security Settings on General Settings Page topic in the Horizon Cloud Administration Guide.</td>
</tr>
</tbody>
</table>

5 In the Primary Server section, specify details about the authentication server.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Host Name / IP Address</td>
<td>Enter the DNS name or the IP address of the authentication server.</td>
</tr>
<tr>
<td>Shared Secret</td>
<td>Enter the secret for communicating with the authentication server. The value must be identical to the server-configured value.</td>
</tr>
<tr>
<td>Authentication Port</td>
<td>Specify the UDP port configured on the authentication server for sending or receiving authentication traffic. The default is 1812.</td>
</tr>
<tr>
<td>Accounting Port</td>
<td>Optionally specify the UDP port configured on the authentication server for sending or receiving accounting traffic. The default is 1813.</td>
</tr>
<tr>
<td>Mechanism</td>
<td>Select the authentication protocol that is supported by the specified authentication server and which you want the deployed pod to use.</td>
</tr>
<tr>
<td>Server Timeout</td>
<td>Specify the number of seconds that the pod should wait for a response from the authentication server. After this number of seconds, a retry is sent if the server does not respond.</td>
</tr>
<tr>
<td>Max Number of Retries</td>
<td>Specify the maximum number of times the pod should retry failed requests to the authentication server.</td>
</tr>
<tr>
<td>Realm Prefix</td>
<td>Optionally provide a string which the system will place at the beginning of the user name when the name is sent to the authentication server. The user account location is called the realm. For example, if the user name is entered as user1 on the login screen and a realm prefix of DOMAIN-A \ was specified here, the system sends DOMAIN-A\user1 to the authentication server. If you do not specify a realm prefix, only the entered user name is sent.</td>
</tr>
<tr>
<td>Realm Suffix</td>
<td>Optionally provide a string which the system will append to the user name when the name is sent to the authentication server. For example, if the user name is entered as user1 on the login screen and a realm suffix of @example.com was specified here, the system sends <a href="mailto:user1@example.com">user1@example.com</a> to the authentication server.</td>
</tr>
</tbody>
</table>
6 (Optional) In the Secondary Server section, optionally specify details about an auxiliary authentication server.

You can configure a secondary authentication server to provide for high availability. Enable the Auxiliary Server toggle and complete the fields as described in Step Primary Server section.

**Validate and Proceed, and then Start the Pod Deployment Process**

After you click Validate & Proceed, the system verifies your specified values. If everything validates, the wizard displays a summary of the information for your review. Then you start the deployment process.

**Procedure**

1. **Click Validate & Proceed.**

   The system validates your specified values, such as:

   - Are the specified address ranges for the to-be-created subnets valid and non-overlapping with other addresses in the selected region within your subscription.
   - Are there enough virtual machine (VM) and cores in your subscription's quota to build out the pod.
   - Are any uploaded certificate files in the correct PEM format.
   - If you selected to use an existing management subnet, does it have the Microsoft.SQL service endpoint enabled on that subnet?

   **Important** Starting in this release, new pod deployments require the Microsoft.SQL service endpoint enabled on the management subnet to support use of the pod's Azure Postgres database. If you see a validation error similar to the following screenshot that lists your selected subnet, it means the pre-existing management subnet that you selected in the wizard is missing that Microsoft.SQL service endpoint configured on it. At this point, you can log in to the Microsoft Azure portal and enable the Microsoft.SQL service endpoint on the subnet. Then you can resubmit the wizard to deploy the pod. For some details on how to enable that endpoint, see Optionally Create the Pod's Required Subnets on your VNet in Microsoft Azure.

   ![Pod configuration validation error](image)

   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.

Deploying your first pod can take up to an hour. Until the pod is successfully deployed, a progress icon is displayed in the Administration Console’s Getting Started screen. You might need to refresh the screen in your browser to see the progress. The browser-based user interface can time out after approximately 30 minutes and ask you to log back in.

**Important** The pod's pending stage typically lasts up to ten minutes. However, when deploying a pod in Microsoft Azure China cloud, the overall deployment process can take up to seven (7) hours to complete. The process is subject to geographic network issues that can cause slow download speeds as the binaries are downloaded from the cloud control plane.

If the pod has not moved from Pending to Downloading state after 20 minutes, and you are not deploying into Microsoft Azure China, the system automatically puts the pod into Error state and displays a message that states the pod cannot connect to the cloud services and to check the networking connectivity in your Microsoft Azure environment.

If the display shows the pod is in Error state because the deployed jump box VM could not download the binaries it needs from the cloud service, there is likely an issue with your environment's network configuration. For example, the VNet's configured DNS might not be resolving internal or external names or the required outbound ports are not open or are blocked by your firewall. Sometimes there is a temporary loss of connectivity to packages.microsoft.com site used to download the Microsoft Azure Command Line Interface software. You can run some tests to verify if your environment's networking is configured properly for the pod's requirements. See **Troubleshooting If You Encounter Pod Deployment or First-Time Domain Bind Issues**.

Throughout the pod deployment process, the Getting Started page's Capacity section indicates the various stages the process goes through (pending, downloading, building, connecting, and so on).

The following table gives some approximate sample durations for the stages in building the pod.

**Important** The actual durations you experience in your deployment's progress will vary depending on the network latencies that exist at the time.

<table>
<thead>
<tr>
<th>Stage</th>
<th>Sample duration</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pending</td>
<td>10 minutes</td>
</tr>
<tr>
<td>Downloading</td>
<td>10 minutes</td>
</tr>
</tbody>
</table>
When the pod is successfully deployed:

- Horizon Cloud sends a notification email to the account owner that is identified in the corresponding Horizon Cloud customer account record. The email states the pod onboarding is complete.
- A green checkmark is displayed in the Getting Started screen next to a message saying the pod was added and about completing the domain join process.

![Microsoft Azure, 1 Pod Complete](image)

**Note** At this point, because your Active Directory domain is not yet registered with the pod, a **Delete** button is displayed in the page. 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 that **Delete** button to delete the artifacts that were deployed. When the screen indicates the pod is successfully deleted, you can start the process over by clicking **Add** again.

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 Horizon Cloud Getting Started wizard and complete the required task of registering an Active Directory domain. Registering Active Directory is the next required step. After registering the domain, you continue management of this pod in the Administration Console. See the Getting Started chapter of *Horizon Cloud Administration Guide*. After registering the Active Directory domain, follow the Getting Started wizard to see which task to complete next.
If you specified a gateway configuration on the pod, before your end users can access their desktops or remote applications, you must set up the appropriate CNAME records in your DNS server according to the type of gateway you specified.

- For an external gateway enabled with a public IP address, map the FQDN that you entered in the deployment wizard to the gateway's Azure load balancer resource's auto-generated public FQDN. Your DNS server record maps that load balancer's auto-generated public FQDN with the FQDN that your end users will use, and which is used in the uploaded certificate. The following code line demonstrates an example.

  `ourApps.ourOrg.example.com   vwm-hcs-podID-uag.region.cloudapp.azure.com`

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

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

If you specified both the external and internal gateway configurations and used the same FQDN for both, you must configure the routing of the incoming end-user client traffic to the appropriate load balancer resource in the gateways' resource groups. In this scenario, you need to set up the routing so that client traffic from the Internet is routed to the external gateway's Azure load balancer resource and client traffic from your intranet is routed to the internal gateway's Azure load balancer resource.

See the *Horizon Cloud Administration Guide* for the steps to locate the load balancer information in the pod's details page.

If you specified RADIUS two-factor authentication for the pod's gateway configurations, you must complete the following tasks.

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

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

For information on how to obtain those IP addresses, see the Administration Guide.

If you specified to have the pod deployment process create a VMware Identity Manager™ tenant, additional steps are required to complete the integration of your pod with that tenant. After you complete the required task of registering an Active Directory domain, see the information in the Integrate a Horizon Cloud Pod with a VMware Identity Manager™ Environment topic in the Horizon Cloud Administration Guide.

Troubleshooting If You Encounter Pod Deployment or First-Time Domain Bind Issues

If your environment's networking is not configured properly for use with the Horizon Cloud pod in Microsoft Azure, the process to build out the pod can get stuck in PENDING state or the post-deployment action to domain bind to your Active Directory environment might fail. The two most common network-related causes are failing to open the required outbound ports and failing to enable the DNS to resolve both internal and external addresses. By following the troubleshooting steps here, you can run some tests to verify the required outbound ports are open and the DNS can resolve both internal and external addresses.

The overall networking requirements for successfully deploying a pod are stated in the prerequisites checklist document, located at this PDF link and described in Configure the Virtual Network's DNS Server and DNS Requirements for a Horizon Cloud Pod in Microsoft Azure. If your environment's networking does not meet those requirements, you will encounter one or both of these two issues:

<table>
<thead>
<tr>
<th>Issues</th>
<th>Common causes</th>
</tr>
</thead>
<tbody>
<tr>
<td>■ The Getting Started page shows the pod in pending state and it never proceeds to connecting state. Usually a pod is in pending state for about 10 minutes (except when deploying a pod into Microsoft Azure China cloud, which takes longer).</td>
<td></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>■ 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>■ 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>
<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 Virtual Network's DNS Server.</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 Administration Console’s Delete action to delete the pod, unexpected results might occur. When deleting a pod, the system checks the pod’s subnets to verify that everything connected to the subnets belongs to the pod itself (according to the pod’s ID). If the system determines additional VMs, VM disks, IPs, or other artifacts are connected to the pod’s subnets, the system cannot cleanly delete the pod.

For details about running the troubleshooting tests, see the following sections.

**Important**  If you are directing all traffic out through your on-premises network and only allowing authenticated traffic to pass, but you did not provide values for using a proxy in the pod deployment wizard, even though all of these manual tests will succeed, the traffic sent by an unauthenticated source, the jump box, will fail. The symptom of this situation is the pod deployment is stuck in pending state. If you are in this situation, you must delete the pod from the Getting Started page, re-run the pod deployment wizard, and specify the required proxy information.

**Procedure**

1. **Create an SSH Key Pair**
   You need an SSH key pair to authenticate to the test Linux VM that you will be deploying into your Microsoft Azure subscription. You create the key pair on the system you will use to SSH connect to the test VM. This step is optional if you already have a key pair on that system.

2. **Create the Test Virtual Machine in Your Microsoft Azure Subscription**
   You will use a test Linux virtual machine (VM) in your Microsoft Azure environment to run the tests that check the network connectivity that is configured for your Horizon Cloud pod.

3. **Use SSH to Connect to the Test VM**
   Make a SSH (Secure Shell) connection to the test VM so that you can run the network connectivity tests in your Microsoft Azure environment.
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.

Delete the Test VM After You Complete the Tests

When you have finished the tests to check your Microsoft Azure network configuration and no longer need the test VM, you should delete it and all of its related artifacts from your Microsoft Azure environment.

Create an SSH Key Pair

You need an SSH key pair to authenticate to the test Linux VM that you will be deploying into your Microsoft Azure subscription. You create the key pair on the system you will use to SSH connect to the test VM. This step is optional if you already have a key pair on that system.

You can use either a Microsoft Windows or a Linux system to create this SSH key pair. The steps for both types of systems are described here. Choose the steps most suitable for your situation.

Create an SSH Key Pair on a Microsoft Windows System

Use these steps when you will be using a Microsoft Windows system to SSH connect to the test Linux VM that you will be deploying into your Microsoft Azure subscription.

When you create the test VM in Microsoft Azure, you will use the contents of the generated public key file. If you already have an existing SSH key pair on the Microsoft Windows system that you will use to connect with the test VM, then you can skip this step and proceed with creating the test VM, as described in Create the Test Virtual Machine in Your Microsoft Azure Subscription.

By following these steps, you generate the SSH key pair, copy the public key file’s contents so you can use it when creating the test VM, and load the private key into the PuTTY Pageant tool. Pageant is an SSH authentication agent that can hold your private keys in memory. By holding the private key in memory, the private key is automatically applied against any SSH session from that Microsoft Windows system, making it easier to use.

Prerequisites

A Microsoft Windows system does not have SSH key-pair software installed on it by default. Verify that SSH key-pair generating software is installed on the system you are planning to use. You can use any SSH key-pair generating software. The steps below describe using the PuTTY software on Microsoft Windows to create the SSH key pair. You can obtain the PuTTY software from www.putty.org. After the installation, the PuTTY suite of tools is available. The following screenshot shows an example of the PuTTY tools in the Start menu.
Procedure

1. On your Microsoft Windows system, launch PuTTYgen (the PuTTY key generator).

On Microsoft Windows 10, the PuTTYgen choice in the Start menu looks like

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 Move your cursor around randomly in the blank area underneath the progress bar. Moving your cursor around in the area adds required randomness into the process.
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 Microsoft Windows 10, the Pageant choice in the Start menu looks like ![Pageant](image). When you click it, the Pageant icon of a computer wearing a hat is loaded into the system tray.

The following screenshot shows the Pageant icon loaded into a Microsoft Windows 10 system tray.
7  Add your private key to Pageant by right-clicking that system tray icon, clicking **Add Key**, and using the file selection window to navigate to and select your saved private key (PPK) file.

![Pageant interface]

**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.

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:

   ```
   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.

   ```
   $ ssh-keygen -t rsa -b 2048
   Generating public/private rsa key pair.
   Enter file in which to save the key (/mts-cm/home/user1/.ssh/id_rsa): mykey
   Enter passphrase (empty for no passphrase):
   Enter same passphrase again:
   ```

   **Note** Using a key passphrase is an optional best practice.

   The private key is saved in the file that you specify and the public key is saved to a file with that same name and a .pub extension. Using the example above of entering `mykey` as the file, the sample output would be:

   ```
   Your identification has been saved in mykey.
   Your public key has been saved in mykey.pub.
   ```

What to do next

Create the test VM by following the steps in Create the Test Virtual Machine in Your Microsoft Azure Subscription.

Create the Test Virtual Machine in Your Microsoft Azure Subscription

You will use a test Linux virtual machine (VM) in your Microsoft Azure environment to run the tests that check the network connectivity that is configured for your Horizon Cloud pod.

**Prerequisites**

Verify that you have the SSH public key that you created as described in Create an SSH Key Pair. You will provide that public key in the VM creation wizard so that the VM will trust SSH connections coming from the system that has the corresponding private key.

Verify you have the name of the virtual network (VNet) that is the same one that you are using to deploy your pod, as described in Configure the Required Virtual Network in Microsoft Azure.
If you tried to deploy a pod and the deployment process failed at some point, the process might already have created the pod’s management subnet in the VNet.

- If so, it is recommended you deploy the test VM on to that subnet. To identify if the pod’s management subnet exists on the VNet, log in to the Microsoft Azure portal, navigate to that VNet, and examine the list of subnets it has. You can navigate to the VNet using the Virtual networks choice in the left navigation bar. The pod’s management subnet will have a name in the pattern vmw-hcs-podID-net-management, where podID is the pod’s UUID.

- If the pod deployment process did not create the pod’s management subnet on the VNet, you can choose any available subnet on the VNet or create a new subnet for the test VM to use.

**Procedure**

1. Log in to the Microsoft Azure portal.

2. From the portal’s left navigation bar, click Virtual machines and then click + Add.


4. Select Resource Manager as the deployment model and click Create.

The Create Virtual Machine wizard opens to the steps for configuring basic settings.

5. Provide the VM’s basic settings and then click OK to move to the next wizard step.

<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Name</td>
<td>Type a name for the VM.</td>
</tr>
<tr>
<td>VM disk type</td>
<td>Keep the default SSD setting.</td>
</tr>
<tr>
<td>User name</td>
<td>Enter a user name that meets the Microsoft Azure user name requirements, as described in the Microsoft documentation here.</td>
</tr>
<tr>
<td><strong>Important</strong></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 SSH public key.</td>
</tr>
<tr>
<td>Option</td>
<td>Description</td>
</tr>
<tr>
<td>---------------</td>
<td>--------------------------------------------------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td><strong>SSH public key</strong></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 ***** BEGIN SSH2 PUBLIC KEY ***** and end with the line ***** END SSH2 PUBLIC KEY ***** from your public key.</td>
</tr>
<tr>
<td><strong>Subscription</strong></td>
<td>Select the same subscription that you are using for your pod.</td>
</tr>
<tr>
<td><strong>Resource group</strong></td>
<td>The recommended choice is to create a new resource group for the test VM and its related artifacts like its disk. Select <strong>Create new</strong> and enter a name for the new resource group. Even though you can use an existing resource group with this test VM, a resource group specific for the test VM is recommended because it is easier to delete the VM and its related artifacts simply by deleting the whole resource group when you are finished running the tests.</td>
</tr>
<tr>
<td><strong>Location</strong></td>
<td>Select the same physical geographic region that you are using for your pod.</td>
</tr>
</tbody>
</table>

![Create virtual machine diagram](image)

**Basics**
- **Name**: HCS-testingVM
- **VM disk type**: SSD
- **User name**: testvmadmin
- **Authentication type**: SSH public key
- **SSH public key**: 
  ```
  ----- BEGIN SSH2 PUBLIC KEY -----
  AAAAB3Nzc1yc2EAAAABJQAAAQE
  ----- END SSH2 PUBLIC KEY -----
  ```
- **Subscription**: HCS
- **Resource group**:
  - Create new
  - Use existing: mytestVMRG
- **Location**: East US

[OK button]
6  In the wizard's size step, click a VM size click and then **Select** to move to the next wizard step.

Because this is expected to be a short-lived VM, used only to complete the verification tests, you can pick any size. However, because smaller sizes usually have lower associated costs in Microsoft Azure, choosing a small size for the test VM is typical. The following screenshot illustrates the example of choosing the D2S_V3 Standard size.

7  In the wizard's settings step, specify the key networking choices for the test VM.

You make three important choices in this wizard step. The following screenshot illustrates these three key items. After setting the three key network options, you can retain all of the other default values.
<table>
<thead>
<tr>
<th>Option</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Virtual network</td>
<td>You must select the same VNet that you are using to deploy your pod. This VNet should be the one that you configured according to the details in the prerequisites checklist and as described in <a href="#">Configure the Required Virtual Network in Microsoft Azure</a>.</td>
</tr>
<tr>
<td>Subnet</td>
<td>If you already tried to deploy the pod and the process failed, the pod's management subnet might have been created in the virtual network. If the subnet is there, it is recommended to select that subnet for this test VM. Click the <strong>Subnet</strong> choice to navigate to the subnets that exist on the selected virtual network. You might have to hover over the subnet to see its full name in the tooltip. This screenshot illustrates hovering over a subnet to see the naming pattern of a pod's management subnet, in the pattern <code>vmw-hcs-podID-net-management</code>. If the pod deployment process did not create the pod's management subnet on the VNet, select the subnet on your VNet that you identified to use for the test VM (as described in the prerequisites above). <strong>Note</strong> If the pod was successfully deployed, but you are troubleshooting domain-join issues, you might select the pod's desktop subnet for the test VM instead of the management subnet, because domain-join operations are used with the desktop images that get connected to that desktop subnet.</td>
</tr>
<tr>
<td>Public IP address</td>
<td>Select this choice so that the created test VM will have a public IP address assigned to it. Having a public IP address enables you to connect to it over the wide area network (WAN). <strong>Note</strong> Using a public IP might not be technically feasible in your networking configuration. If you cannot create the test VM with a public IP, you will need to have network connectivity from your local system to the subnet you selected in the <strong>Subnet</strong> field or will need to connect to some other machine on your network and then inbound connect to the test VM.</td>
</tr>
</tbody>
</table>

8. Click **OK** to move to the wizard's summary step.

9. In the summary step, verify that the key pieces of information (subscription, location, virtual network, and subnet) match the ones that you are using for your pod, and then click **Create**.
Creation of the test VM starts running. Typically, you can see the process running on your Microsoft Azure dashboard, as illustrated in the following screenshot.

Deploying the VM usually takes around five to ten minutes. When the VM is fully deployed, it is in Running state. The following screenshot illustrates the details of a sample test VM.

When you see the test VM is up and running, connect to it. Follow the steps in Use SSH to Connect to the Test VM.

**Use SSH to Connect to the Test VM**

Make a SSH (Secure Shell) connection to the test VM so that you can run the network connectivity tests in your Microsoft Azure environment.

**SSH Connect to the Test VM from a Microsoft Windows System**

You make this connection from the Microsoft Windows system that has the private key that corresponds to the public key you specified when you created the test VM.

**Prerequisites**

Verify you have the test VM’s IP address and the user name you specified when you created the VM.
On a Microsoft Windows system, PuTTY is typically used. To make it easy for PuTTY to load your private key when you start the SSH session, before starting PuTTY, start Pageant as described in Create an SSH Key Pair on a Microsoft Windows System and add the SSH private key to the Pageant key list. The SSH private key must match the public key you specified when creating the test VM. When the private key is loaded into Pageant, the PuTTY SSH session will use that private key automatically.

**Procedure**

1. Launch PuTTY.
   - The PuTTY Configuration window opens.

2. In the PuTTY Configuration window, specify the host name, select SSH, and then click Open.
   - In the PuTTY Configuration window's Host Name field, type a string in the pattern `testvm_username@testvmip` substituting the test VM's user name and IP address for `testvm_username` and `testvmip` in the string.

   **Important** After you click Open, when this is the first time you connect to the test VM, a PuTTY security message displays stating that the server's host key is not cached and displays the server's rsa2 key fingerprint. You can continue to make the connection by either clicking Yes to add the server's host key into PuTTY's cache or No to connect without adding the key to PuTTY's cache. If you suspect the connection might not be going to your test VM, click Cancel to abandon the connection and return to the PuTTY Configuration window to verify your host name entry.

   The following screenshot is an illustration of the window using the sample `testvmadmin@40.121.180.132`
When the SSH connection is established, a command-line window displays that looks similar to the following screenshot.

What to do next

Now that you are connected to the test VM, you can run the tests to check network connectivity within your Microsoft Azure environment. Follow the steps described in Run the Tests to Check Networking in Your Microsoft Azure Environment.
SSH Connect to the Test VM from a Linux System
You make this connection from the Linux system that has the private key that corresponds to the public key you specified when you created the test VM.

Prerequisites
Verify you have the test VM's IP address and the user name you specified when you created the VM.

Procedure
1. Open a bash shell.
2. At the bash shell $ prompt, enter the ssh command as below, substituting the test VM's IP address and the user name for testvmip and testvm_username in the command:

   ```bash
   ssh testvm_username@testvmip
   ```
   
   For example, using the test VM details from the examples in Create the Test Virtual Machine in Your Microsoft Azure Subscription, the sample command would look like:

   ```bash
   ssh testvmadmin@40.121.180.132
   ```

   When the SSH connection is established, a command-line window displays that looks similar to the following screenshot.

   ![SSH Connection Screenshot]

What to do next
Now that you are connected to the test VM, you can run the tests to check network connectivity within your Microsoft Azure environment. Follow the steps described in Run the Tests to Check Networking in Your Microsoft Azure Environment.
Run the Tests to Check Networking in Your Microsoft Azure Environment

You run these tests to verify these two network-related areas are configured properly: that the DNS can resolve both internal and external addresses and that the required outbound ports are open. You run these tests using your test VM.

The pod relies on DNS to resolve both internal and external addresses. The first two tests here check whether the DNS configured in your network environment can resolve known FQDNs for internal and external addresses.

**Important** If you are directing all traffic out through your on-premises network and only allowing authenticated traffic to pass, but you did not provide values for using a proxy in the pod deployment wizard, even though all of these manual tests will succeed, the traffic sent by an unauthenticated source, the jump box, will fail. The symptom of this situation is the pod deployment is stuck in pending state. If you are in this situation, you must delete the pod from the Getting Started page, re-run the pod deployment wizard, and specify the required proxy information.

**Prerequisites**

Before running these tests, verify that you created a test VM in your Microsoft Azure subscription and have an SSH connection to it, as described in Create the Test Virtual Machine in Your Microsoft Azure Subscription and Use SSH to Connect to the Test VM.

Obtain the IP addresses and fully qualified domain names (FQDNs) for servers that are internal to your network that you expect to be reachable from the VNet, such as your Active Directory Domain Controller. You will use this information in the DNS verification test.

**Procedure**

1. Check that DNS is working in your environment to resolve internal FQDNs by using the `dig` command to query a known domain name that is internal to your VNet in Microsoft Azure.

   In the SSH connection window, issue the `dig` command to query the domain name of a server that you know is internal to your network, such as your Active Directory Domain Controller.

   ```
dig internal-domain-name
   ``

   Where `internal-domain-name` is the fully qualified domain name of a server that you know is internal to your network.

   `dig` (Domain Information Groper) is a command-line tool for network troubleshooting. By running this command using an internal host name, the result verifies that your DNS configuration can resolve internal addresses properly. If your DNS configuration can resolve the `internal-domain-name` used in the command, the command output will return the correct IP address associated with that domain name.
For example, assume the VNet is configured with an internal Active Directory server having Active Directory Domain Controller with a DNS entry of skylo.local and an IP address of 192.168.0.15. Issuing `dig skylo.local` would check whether the VNet's DNS configuration can resolve that internal skylo.local server name:

```
testvmadmin@HCS-testingVM:~$ dig skylo.local
;; DiG 9.10.3-P4-Ubuntu <<< skylo.local
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 64899
;; flags: qr aa rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
;; EDNS: version: 0, flags:; udp: 4000
;; QUESTION SECTION:
skylo.local.            IN      A

;; ANSWER SECTION:
skylo.local.            600     IN      A       192.168.0.15

;; Query time: 1 msec
;; SERVER: 192.168.0.15#53(192.168.0.15)
;; WHEN: Mon Mar 26 20:58:01 UTC 2018
;; MSG SIZE  rcvd: 56

testvmadmin@HCS-testingVM:~$
```

The test is successful when the ANSWER SECTION indicates the provided host name was resolved to the IP address that you expect for that host name.

**Note** Sometimes DNS is not 100% reliable, and some requests resolve fine while others fail. If issuing the command fails the first time, run the command for ten to twenty iterations and see whether you get reliable responses each time.

2 Check that DNS is working in your environment to resolve external FQDNs by using the `dig` command to query a known external domain name.

In the SSH connection window, issue the `dig` command to query an external industry-standard domain name, such as `vmware.com` or `microsoft.com`.

```
dig external-domain-name
```

Where `external-domain-name` is a fully qualified domain name that is external to your VNet. For example, issuing `dig vmware.com` would check whether the VNet's DNS configuration could resolve that external name:

```
testvmadmin@HCS-testingVM:~$ dig vmware.com
;; DiG 9.10.3-P4-Ubuntu <<< vmware.com
;; global options: +cmd
;; Got answer:
```
In the above example, the ANSWER SECTION indicates the external domain name `vmware.com` was properly resolved to two IP addresses.

**Note** You can repeat this test using various external domain names, such as `azure.com` or `microsoft.com`, to verify that your DNS can resolve different external names.

If the DNS tests do not work, verify your network configurations and your DNS server. Check that you added your DNS server to your VNet.

**Important** If you find that you need to add your DNS server to your VNet or you have to change the VNet's DNS server configuration, you must restart all VMs that are connected to that VNet for them to pick up the change. If you change the VNet's DNS server configuration and do not restart all of the VMs connected to that VNet, the changes will not propagate correctly on the VNet.

3 Check that the required outbound ports are available by using the `netcat` command.

Horizon Cloud requires some outbound ports to be opened, so that the pod software can be securely downloaded into your Microsoft Azure environment and so that the pod can connect back to the Horizon Cloud control plane. As described in [DNS Requirements for a Horizon Cloud Pod in Microsoft Azure](#), the following outbound TCP ports are required to be open from the pod's management subnet: port 80, 443, and 11371. By running the `netcat` command as indicated in the command below, you can verify that those outbound ports are open as required.

In the SSH connection window, issue the following commands (one per port).

**Note** The command below to test port 11371 specifies `packages.microsoft.com` to test that connection, while the other two lines test the outbound connection to the Horizon Cloud control plane.

```
testvmadmin@HCS-testingVM:$ netcat -v -w 3 cloud.horizon.vmware.com 80
Connection to cloud.horizon.vmware.com 80 port [tcp/http] succeeded!
```
When a port is properly open, the netcat command returns the `succeeded!` line for its test.

If the netcat commands return failures, check your Microsoft Azure network connections, your Network Security Groups in your subscription, and any firewalls you might have in place. Ensure your network configuration meets the DNS, ports, and protocol requirements that the pod needs for deployment, as described in DNS Requirements for a Horizon Cloud Pod in Microsoft Azure.

If the above tests succeed, you will be able to successfully deploy your pod.

**Note**  If you will be configuring optional features for use with your pod, such as Radius two-factor authentication or True SSO, additional ports might be needed for those purposes. You can use the above outbound port testing techniques to verify such ports are properly open.

**What to do next**

When you have completed the test, you should delete the test VM and all of its related artifacts, such as its VM disk, IP address, NIC, from your Microsoft Azure environment. Ideally you would have created a resource group for the test VM and can simply delete that resource group to delete all of the VM's artifacts. Follow the steps in Delete the Test VM After You Complete the Tests.

**Important**  If you do not delete all of the test VM's artifacts from your Microsoft Azure environment and you connected the VM to one of the pod's subnets, if you later try to delete the pod using the Administration Console, 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 using the Administration Console, the system might not be able to fully delete the pod due to those remaining connected artifacts. By default, when you use the Delete action to delete a pod, Horizon Cloud deletes those resource groups and subnets that it created for the pod. Microsoft Azure will prevent deletion of subnets that are still in use. If your test VM's artifacts are connected to the pod's subnets, then those subnets cannot be deleted and the pod deletion will be incomplete. To prevent this situation, ensure all of the test VM's artifacts are deleted after you have successfully deployed your pod.
Procedure

1. Log in to the Microsoft Azure portal.

2. Use one of the following methods to delete the test VM, depending on how you deployed it.
   - If you deployed the test VM into its own resource group and you are not using that group for any other purpose, you can delete the entire resource group.

   **Caution** To avoid inadvertently deleting other items, make sure that the resource group contains only your test VM and its associated objects such as its disk and network adapters before deleting the resource group.

   a. In the portal's left hand navigation, click **Resource groups** and search for the test VM's resource group.

   b. Click the resource group's name to see the items in that resource group.

   c. Click **Delete resource group**. In the confirmation message, type in the resource group's name and then click **Delete**.
If you need to delete the test VM without deleting an entire resource group, you can use the portal's search box to search for the test VM's name. The results of this search will list the VM and all of its associated objects (disk, network interfaces, public IP address, and so on). Then delete each object individually.
Congratulations on deploying your first ever Horizon Cloud pod!

The Administration 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.