Install an agent on-premises

NetApp Console setup and administration

NetApp
January 27, 2026

Table of Contents

Install an agent on-premises . . . 1
- Manually install a Console agent on-premises . . . 1
  - Prepare to install the Console agent . . . 1
  - Manually install a Console agent . . . 15
  - Register the Console agent with NetApp Console . . . 21
  - Provide cloud provider credentials to NetApp Console . . . 21
- Install a Console agent on-premises using VCenter . . . 22
  - Prepare to install the Console agent . . . 23
  - Install a Console agent in your VCenter environment . . . 34
  - Register the Console agent with NetApp Console . . . 36
  - Add cloud provider credentials to the Console . . . 36
- Ports for the on-premises Console agent . . . 37

Install an agent on-premises

Manually install a Console agent on-premises

Install a Console agent on-premises and then log in and set it up to work with your Console organization.

If you are a VMWare user, you can use an OVA to install a Console agent in your VCenter. Learn more about installing an agent in a VCenter.

Before you install, you'll need to ensure your host (VM or Linux host) meets requirements and ensure that the Console agent will have outbound access to the internet as well as targeted networks. If you plan to NetApp data services, or cloud storage options such as Cloud Volumes ONTAP, you'll need to create credentials in your cloud provider to add to the Console so that the Console agent can perform actions in the cloud on your behalf.

Prepare to install the Console agent

Before you install a Console agent, you should ensure you have a host machine that meets installation requirements. You'll also need to work with your network administrator to ensure that the Console agent has outbound access to required endpoints and connections to targeted networks.

Review Console agent host requirements

Run the Console agent on a x86 host that meets operating system, RAM, and port requirements. Ensure that your host meets these requirements before you install the Console agent.

The Console agent reserves the UID and GID range of 19000 to 19200. This range is fixed and cannot be modified. If any third-party software on your host is using UIDs or GIDs within this range, the agent installation will fail. NetApp recommends using a host that is free of third-party software to avoid conflicts.

Dedicated host

The Console agent requires a dedicated host. Any architecture is supported if it meets these size requirements:

- CPU: 8 cores or 8 vCPUs
- RAM: 32 GB
- Disk space: 165 GB is recommended for the host, with the following partition requirements:
  - `/opt`: 120 GiB of space must be available

    The agent uses `/opt` to install the `/opt/application/netapp` directory and its contents.

  - `/var`: 40 GiB of space must be available

    The Console agent requires this space in `/var` because Podman or Docker are architected to create the containers within this directory. Specifically, they will create containers in the `/var/lib/containers/storage` directory and `/var/lib/docker` for Docker. External mounts or symlinks do not work for this space.

Hypervisor

A bare metal or hosted hypervisor that is certified to run a supported operating system is required.

Operating system and container requirements

The Console agent is supported with the following operating systems when using the Console in standard mode or restricted mode. A container orchestration tool is required before you install the agent.

Operating system	Supported OS versions	Supported agent versions	Required container tool	SELinux
Red Hat Enterprise Linux				
	9.6 • English language versions only. • The host must be registered with Red Hat Subscription Management. If it's not registered, the host can't access repositories to update required 3rd-party software during agent installation.	4.0.0 or later with the Console in standard mode or restricted mode	Podman version 5.4.0 with podman-compose 1.5.0. View Podman configuration requirements.	Supported in enforcing mode or permissive mode

Operating system	Supported OS versions	Supported agent versions	Required container tool	SELinux
	9.1 to 9.4 • English language versions only. • The host must be registered with Red Hat Subscription Management. If it's not registered, the host can't access repositories to update required 3rd-party software during agent installation.	3.9.50 or later with the Console in standard mode or restricted mode	Podman version 4.9.4 with podman-compose 1.5.0. View Podman configuration requirements.	Supported in enforcing mode or permissive mode
	8.6 to 8.10 • English language versions only. • The host must be registered with Red Hat Subscription Management. If it's not registered, the host can't access repositories to update required 3rd-party software during agent installation.	3.9.50 or later with the Console in standard mode or restricted mode	Podman version 4.6.1 or 4.9.4 with podman-compose 1.0.6. View Podman configuration requirements.	Supported in enforcing mode or permissive mode
Ubuntu				
	24.04 LTS	3.9.45 or later with the NetApp Console in standard mode or restricted mode	Docker Engine 23.06 to 28.0.0.	Not supported

Operating system	Supported OS versions	Supported agent versions	Required container tool	SELinux
	22.04 LTS	3.9.50 or later	Docker Engine 23.0.6 to 28.0.0.	Not supported

Set up network access for the Console agent

Set up network access to ensure the Console agent can manage resources. It needs connections to target networks and outbound internet access to specific endpoints.

Connections to target networks

The Console agent requires a network connection to the location where you're planning to create and manage systems. For example, the network where you plan to create Cloud Volumes ONTAP systems or a storage system in your on-premises environment.

Outbound internet access

The network location where you deploy the Console agent must have an outbound internet connection to contact specific endpoints.

Endpoints contacted from computers when using the web-based NetApp Console

Computers that access the Console from a web browser must have the ability to contact several endpoints. You'll need to use the Console to set up the Console agent and for day-to-day use of the Console.

Prepare networking for the NetApp console.

Endpoints contacted from the Console agent

The Console agent requires outbound internet access to contact the following endpoints to manage resources and processes within your public cloud environment for day-to-day operations.

The endpoints listed below are all CNAME entries.

A Console agent installed on your premises cannot manage resources in Google Cloud. If you want to manage Google Cloud resources, you need to install an agent in Google Cloud.

AWS

When the Console agent is installed on-premises, it needs network access to the following AWS endpoints in order to manage NetApp systems (such as Cloud Volumes ONTAP) deployed in AWS.

Endpoints contacted from the Console agent

The Console agent requires outbound internet access to contact the following endpoints to manage resources and processes within your public cloud environment for day-to-day operations.

The endpoints listed below are all CNAME entries.

Endpoints	Purpose
AWS services (amazonaws.com): • CloudFormation • Elastic Compute Cloud (EC2) • Identity and Access Management (IAM) • Key Management Service (KMS) • Security Token Service (STS) • Simple Storage Service (S3)	To manage AWS resources. The endpoint depends on your AWS region. Refer to AWS documentation for details
Amazon FsX for NetApp ONTAP: • api.workloads.netapp.com	The web-based console contacts this endpoint to interact with the Workload Factory APIs to manage and operate FSx for ONTAP based workloads.
https://mysupport.netapp.com	To obtain licensing information and to send AutoSupport messages to NetApp support.
https://signin.b2c.netapp.com	To update NetApp Support Site (NSS) credentials or to add new NSS credentials to the NetApp Console.
https://support.netapp.com	To obtain licensing information and to send AutoSupport messages to NetApp support as well as to receive software updates for Cloud Volumes ONTAP.
https://api.bluexp.netapp.com https://netapp-cloud-account.auth0.com https://netapp-cloud-account.us.auth0.com https://console.netapp.com https://components.console.bluexp.netapp.com https://cdn.auth0.com	To provide features and services within the NetApp Console.

Endpoints	Purpose
https://bluexpinfraprod.eastus2.data.azurecr.io https://bluexpinfraprod.azurecr.io	To obtain images for Console agent upgrades. • When you deploy a new agent, the validation check tests connectivity to current endpoints. If you use previous endpoints, the validation check fails. To avoid this failure, skip the validation check. Although the previous endpoints are still supported, NetApp recommends updating your firewall rules to the current endpoints as soon as possible. Learn how to update your endpoint list. • When you update to the current endpoints in your firewall, your existing agents will continue to work.

Azure

When the Console agent is installed on-premises, it needs network access to the following Azure endpoints in order to manage NetApp systems (such as Cloud Volumes ONTAP) deployed in Azure.

Endpoints	Purpose
https://management.azure.com https://login.microsoftonline.com https://blob.core.windows.net https://core.windows.net	To manage resources in Azure public regions.
https://management.chinacloudapi.cn https://login.chinacloudapi.cn https://blob.core.chinacloudapi.cn https://core.chinacloudapi.cn	To manage resources in Azure China regions.
https://mysupport.netapp.com	To obtain licensing information and to send AutoSupport messages to NetApp support.
https://signin.b2c.netapp.com	To update NetApp Support Site (NSS) credentials or to add new NSS credentials to the NetApp Console.
https://support.netapp.com	To obtain licensing information and to send AutoSupport messages to NetApp support as well as to receive software updates for Cloud Volumes ONTAP.

Endpoints	Purpose
https://api.bluexp.netapp.com https://netapp-cloud-account.auth0.com https://netapp-cloud-account.us.auth0.com https://console.netapp.com https://components.console.bluexp.netapp.com https://cdn.auth0.com	To provide features and services within the NetApp Console.
https://bluexpinfraprod.eastus2.data.azurecr.io https://bluexpinfraprod.azurecr.io	To obtain images for Console agent upgrades. • When you deploy a new agent, the validation check tests connectivity to current endpoints. If you use previous endpoints, the validation check fails. To avoid this failure, skip the validation check. Although the previous endpoints are still supported, NetApp recommends updating your firewall rules to the current endpoints as soon as possible. Learn how to update your endpoint list. • When you update to the current endpoints in your firewall, your existing agents will continue to work.

Proxy server

NetApp supports both explicit and transparent proxy configurations. If you are using a transparent proxy, you only need to provide the certificate for the proxy server. If you are using an explicit proxy, you'll also need the IP address and credentials.

- IP address
- Credentials
- HTTPS certificate

Ports

There's no incoming traffic to the Console agent, unless you initiate it or if it is used as a proxy to send AutoSupport messages from Cloud Volumes ONTAP to NetApp Support.

- HTTP (80) and HTTPS (443) provide access to the local UI, which you'll use in rare circumstances.
- SSH (22) is only needed if you need to connect to the host for troubleshooting.
- Inbound connections over port 3128 are required if you deploy Cloud Volumes ONTAP systems in a subnet where an outbound internet connection isn't available.

  If Cloud Volumes ONTAP systems don't have an outbound internet connection to send AutoSupport messages, the Console automatically configures those systems to use a proxy server that's included with the Console agent. The only requirement is to ensure that the Console agent's security group allows inbound connections over port 3128. You'll need to open this port after you deploy the Console agent.

Enable NTP

If you're planning to use NetApp Data Classification to scan your corporate data sources, you should enable a Network Time Protocol (NTP) service on both the Console agent and the NetApp Data Classification system so that the time is synchronized between the systems. Learn more about NetApp Data classification

Create Console agent cloud permissions for AWS or Azure

If you want to use NetApp data services in AWS or Azure with an on-premises Console agent, then you need to set up permissions in your cloud provider and then you add the credentials to the Console agent after you install it.

You must install the Console agent in Google Cloud to manage any resources that reside there.

AWS

When the Console agent is installed on-premises, you need to provide the Console with AWS permissions by adding access keys for an IAM user who has the required permissions.

You must use this authentication method if the Console agent is installed on-premises. You can't use an IAM role.

Steps

1. Log in to the AWS console and navigate to the IAM service.
2. Create a policy:
   a. Select **Policies > Create policy**.
   b. Select **JSON** and copy and paste the contents of the IAM policy for the Console agent.
   c. Finish the remaining steps to create the policy.

      Depending on the NetApp data services that you're planning to use, you might need to create a second policy.

      For standard regions, the permissions are spread across two policies. Two policies are required due to a maximum character size limit for managed policies in AWS. Learn more about IAM policies for the Console agent.
3. Attach the policies to an IAM user.
   - AWS Documentation: Creating IAM Roles
   - AWS Documentation: Adding and Removing IAM Policies
4. Ensure that the user has an access key that you can add to the NetApp Console after you install the Console agent.

Result

You should now have access keys for an IAM user who has the required permissions. After you install the Console agent, associate these credentials with the Console agent from the Console.

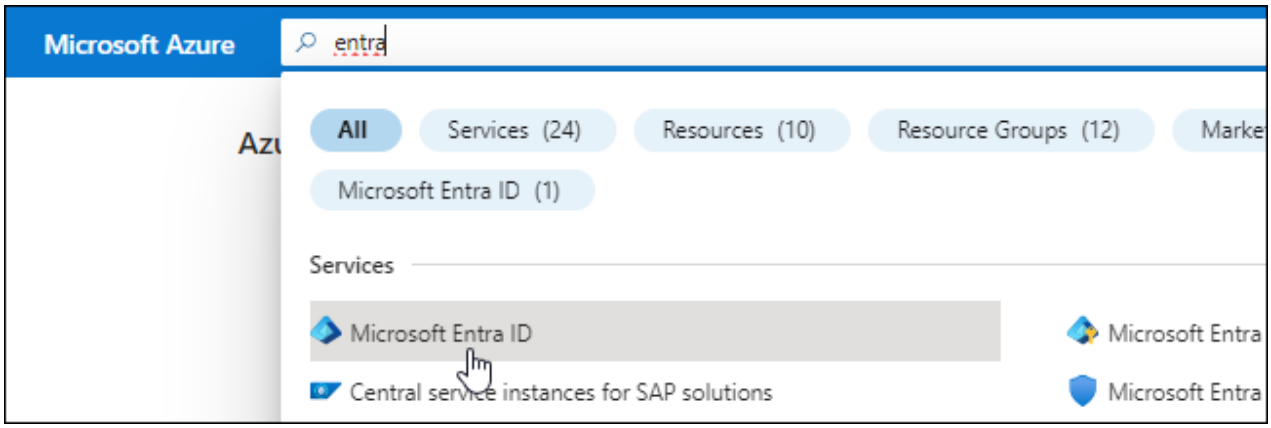
Azure

When the Console agent is installed on-premises, you need to provide the Console agent with Azure permissions by setting up a service principal in Microsoft Entra ID and obtaining the Azure credentials that the Console agent needs.

Create a Microsoft Entra application for role-based access control

1. Ensure that you have permissions in Azure to create an Active Directory application and to assign the application to a role.

   For details, refer to Microsoft Azure Documentation: Required permissions
2. From the Azure portal, open the **Microsoft Entra ID** service.



3. In the menu, select **App registrations**.
4. Select **New registration**.
5. Specify details about the application:
   - **Name**: Enter a name for the application.
   - **Account type**: Select an account type (any will work with the NetApp Console).
   - **Redirect URI**: You can leave this field blank.
6. Select **Register**.

   You've created the AD application and service principal.

Assign the application to a role

1. Create a custom role:

   Note that you can create an Azure custom role using the Azure portal, Azure PowerShell, Azure CLI, or REST API. The following steps show how to create the role using the Azure CLI. If you would prefer to use a different method, refer to Azure documentation

   a. Copy the contents of the custom role permissions for the Console agent and save them in a JSON file.

   b. Modify the JSON file by adding Azure subscription IDs to the assignable scope.

      You should add the ID for each Azure subscription from which users will create Cloud Volumes ONTAP systems.

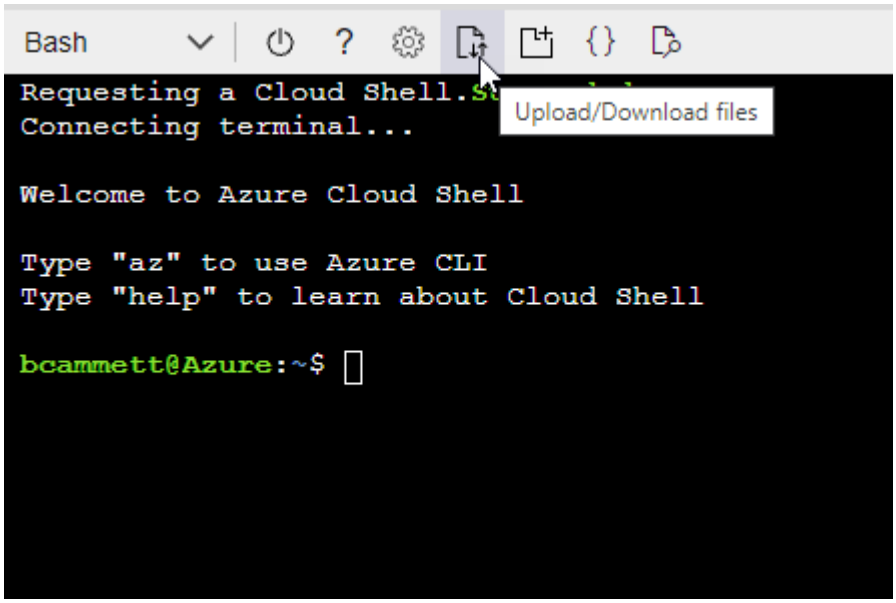
      Example

      ```
      "AssignableScopes": [
      "/subscriptions/d333af45-0d07-4154-943d-c25fbzzzzzzz",
      "/subscriptions/54b91999-b3e6-4599-908e-416e0zzzzzzz",
      "/subscriptions/398e471c-3b42-4ae7-9b59-ce5bbzzzzzzz"
      ]
      ```

   c. Use the JSON file to create a custom role in Azure.

      The following steps describe how to create the role by using Bash in Azure Cloud Shell.

- Start Azure Cloud Shell and choose the Bash environment.
- Upload the JSON file.

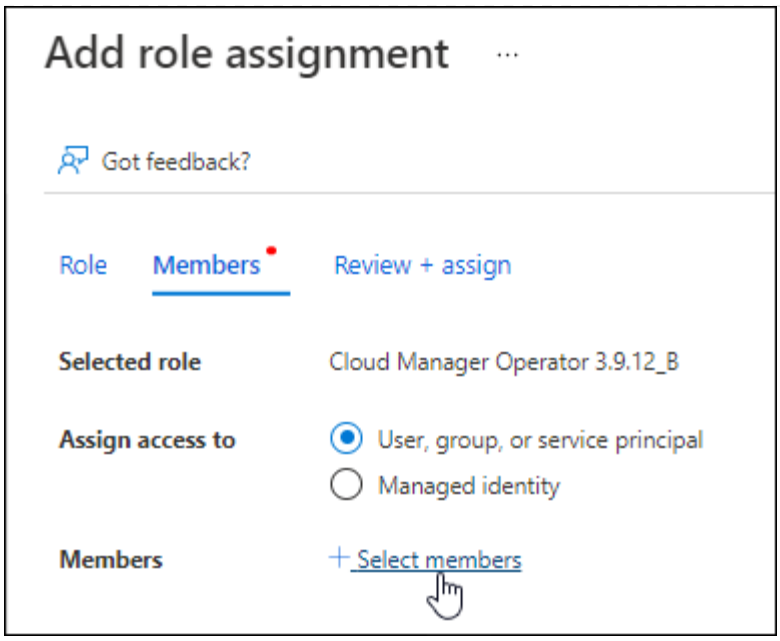


- Use the Azure CLI to create the custom role:

```
az role definition create --role-definition agent_Policy.json
```

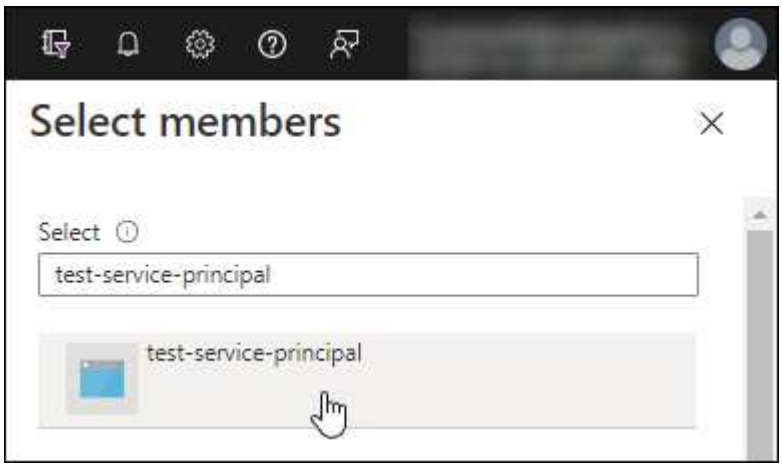
You should now have a custom role called Console Operator that you can assign to the Console agent virtual machine.

2. Assign the application to the role:
   a. From the Azure portal, open the **Subscriptions** service.
   b. Select the subscription.
   c. Select **Access control (IAM) > Add > Add role assignment**.
   d. In the **Role** tab, select the **Console Operator** role and select **Next**.
   e. In the **Members** tab, complete the following steps:
      - Keep **User, group, or service principal** selected.
      - Select **Select members**.



- Search for the name of the application.

  Here's an example:



- Select the application and select **Select**.
- Select **Next**.

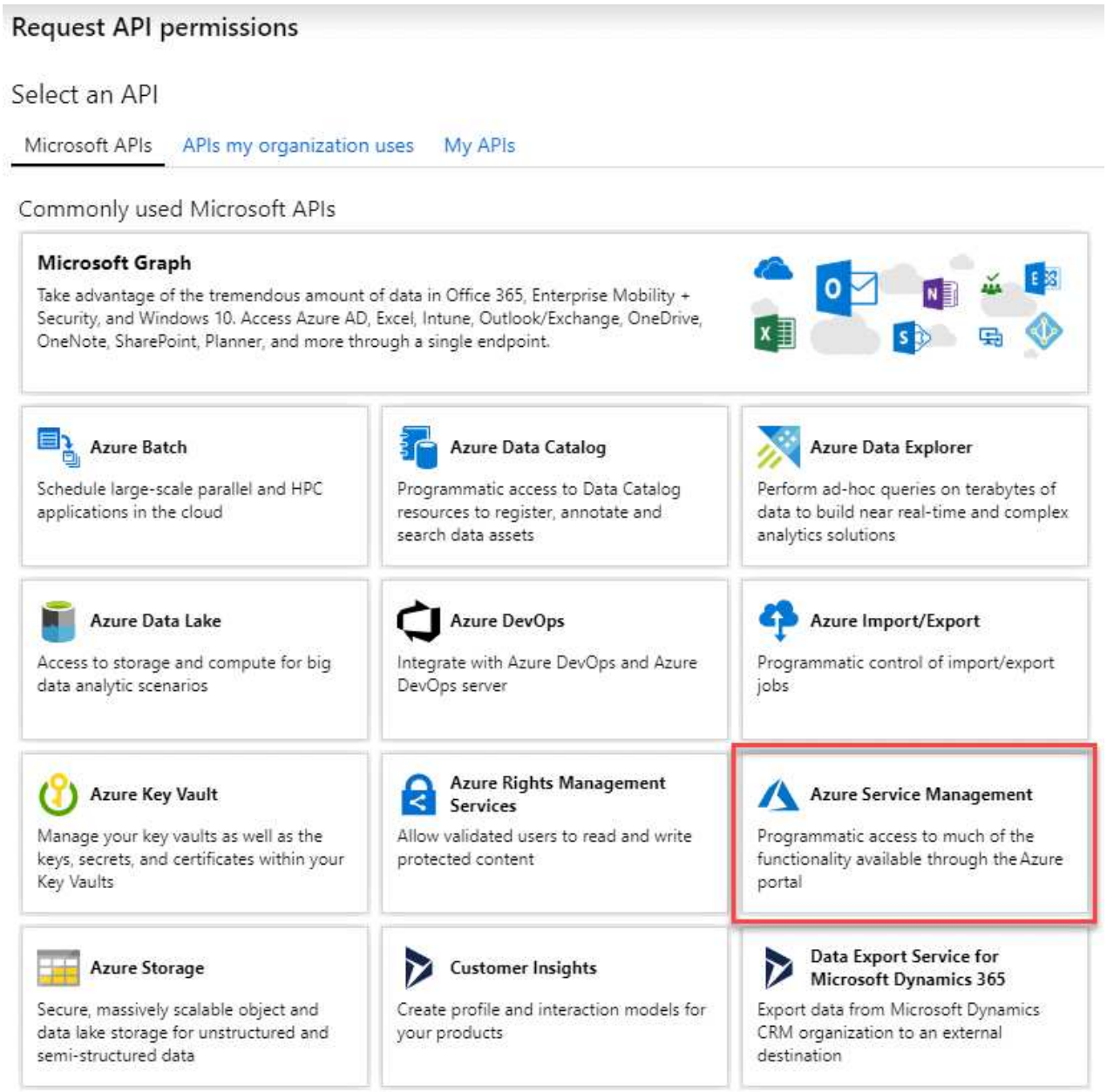
f. Select **Review + assign**.

The service principal now has the required Azure permissions to deploy the Console agent.

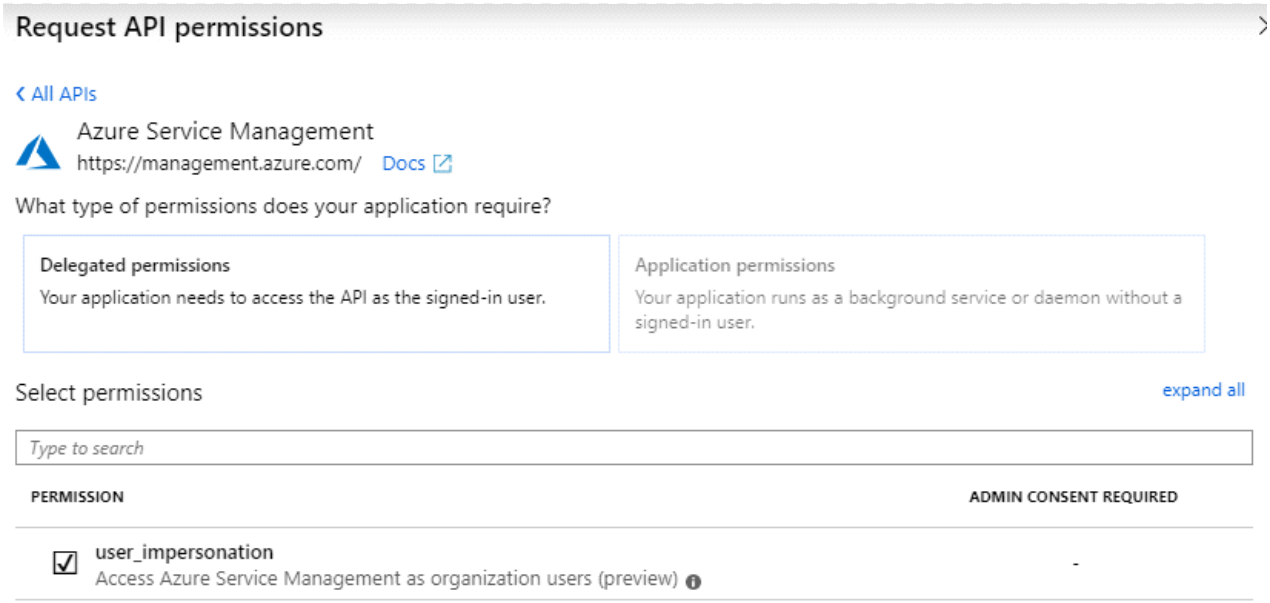
If you want to deploy Cloud Volumes ONTAP from multiple Azure subscriptions, then you must bind the service principal to each of those subscriptions. In the NetApp Console, you can select the subscription that you want to use when deploying Cloud Volumes ONTAP.

Add Windows Azure Service Management API permissions

1. In the **Microsoft Entra ID** service, select **App registrations** and select the application.
2. Select **API permissions > Add a permission**.
3. Under **Microsoft APIs**, select **Azure Service Management**.

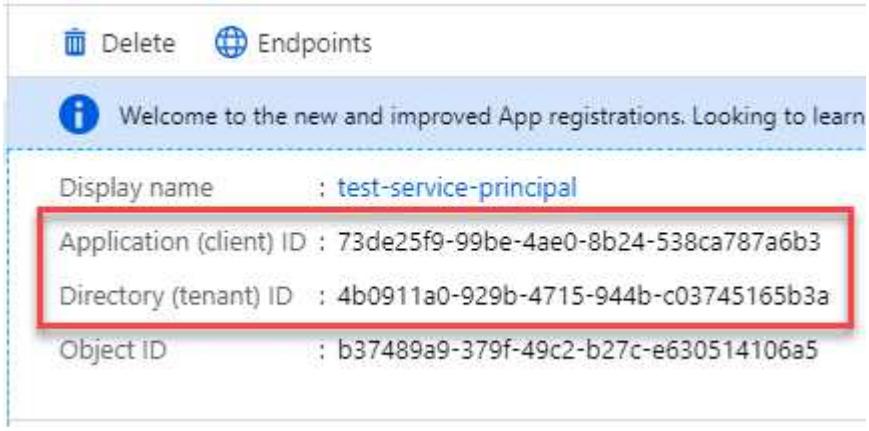


4. Select **Access Azure Service Management as organization users** and then select **Add permissions**.



Get the application ID and directory ID for the application

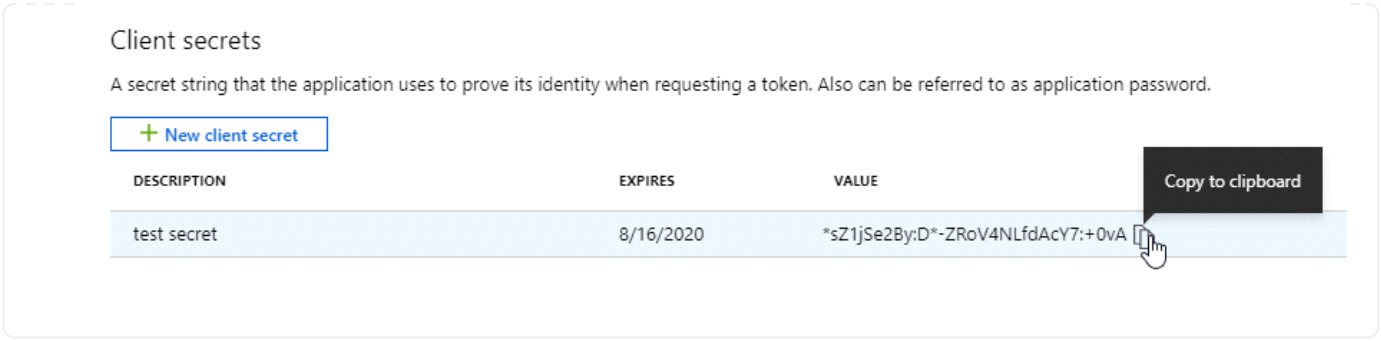
1. In the **Microsoft Entra ID** service, select **App registrations** and select the application.
2. Copy the **Application (client) ID** and the **Directory (tenant) ID**.



When you add the Azure account to the Console, you need to provide the application (client) ID and the directory (tenant) ID for the application. The Console uses the IDs to programmatically sign in.

Create a client secret

1. Open the **Microsoft Entra ID** service.
2. Select **App registrations** and select your application.
3. Select **Certificates & secrets > New client secret**.
4. Provide a description of the secret and a duration.
5. Select **Add**.
6. Copy the value of the client secret.



Manually install a Console agent

When you manually install a Console agent, you need to prepare your machine environment so that it meets requirements. You'll need an Linux machine and you'll need to install Podman or Docker, depending on your Linux operating system.

Install Podman or Docker Engine

Depending on your operating system, either Podman or Docker Engine is required before installing the agent.

- Podman is required for Red Hat Enterprise Linux 8 and 9.

  View the supported Podman versions.

- Docker Engine is required for Ubuntu.

  View the supported Docker Engine versions.

Example 1. Steps

Podman

Follow these steps to install and configure Podman:

- Enable and start the podman.socket service
- Install python3
- Install the podman-compose package version 1.0.6
- Add podman-compose to the PATH environment variable
- If using Red Hat Enterprise Linux, verify that your Podman version is using Netavark Aardvark DNS instead of CNI

Adjust the aardvark-dns port (default: 53) after installing the agent to avoid DNS port conflicts. Follow the instructions to configure the port.

Steps

1. Remove the podman-docker package if it's installed on the host.

```
dnf remove podman-docker
rm /var/run/docker.sock
```

2. Install Podman.

   You can obtain Podman from official Red Hat Enterprise Linux repositories.

   a. For Red Hat Enterprise Linux 9.6:

   ```
   sudo dnf install podman-5:<version>
   ```

   Where <version> is the supported version of Podman that you're installing. View the supported Podman versions.

   b. For Red Hat Enterprise Linux 9.1 to 9.4:

   ```
   sudo dnf install podman-4:<version>
   ```

   Where <version> is the supported version of Podman that you're installing. View the supported Podman versions.

   c. For Red Hat Enterprise Linux 8:

   ```
   sudo dnf install podman-4:<version>
   ```

   Where <version> is the supported version of Podman that you're installing. View the supported Podman versions.

3. Enable and start the podman.socket service.

```
sudo systemctl enable --now podman.socket
```

4. Install python3.

```
sudo dnf install python3
```

5. Install the EPEL repository package if it's not already available on your system.

   This step is required because podman-compose is available from the Extra Packages for Enterprise Linux (EPEL) repository.

6. If using Red Hat Enterprise 9:

   a. Install the EPEL repository package.

```
sudo dnf install https://dl.fedoraproject.org/pub/epel/epel-release-latest-9.noarch.rpm
```

   a. Install podman-compose package 1.5.0.

```
sudo dnf install podman-compose-1.5.0
```

7. If using Red Hat Enterprise Linux 8:

   a. Install the EPEL repository package.

```
sudo dnf install https://dl.fedoraproject.org/pub/epel/epel-release-latest-8.noarch.rpm
```

   b. Install podman-compose package 1.0.6.

```
sudo dnf install podman-compose-1.0.6
```

   Using the `dnf install` command meets the requirement for adding podman-compose to the PATH environment variable. The installation command adds podman-compose to /usr/bin, which is already included in the `secure_path` option on the host.

   c. If using Red Hat Enterprise Linux 8, verify that your Podman version is using NetAvark with Aardvark DNS instead of CNI.

      i. Check to see if your networkBackend is set to CNI by running the following command:

```
podman info | grep networkBackend
```

ii. If the networkBackend is set to `CNI`, you'll need to change it to `netavark`.

iii. Install `netavark` and `aardvark-dns` using the following command:

```
dnf install aardvark-dns netavark
```

iv. Open the `/etc/containers/containers.conf` file and modify the network_backend option to use "netavark" instead of "cni".

If `/etc/containers/containers.conf` doesn't exist, make the configuration changes to `/usr/share/containers/containers.conf`.

v. Restart podman.

```
systemctl restart podman
```

vi. Confirm networkBackend is now changed to "netavark" using the following command:

```
podman info | grep networkBackend
```

Docker Engine

Follow the documentation from Docker to install Docker Engine.

Steps

1. View installation instructions from Docker

   Follow the steps to install a supported Docker Engine version. Do not install the latest version, as it is unsupported by the Console.

2. Verify that Docker is enabled and running.

```
sudo systemctl enable docker && sudo systemctl start docker
```

Install the Console agent manually

Download and install the Console agent software on an existing Linux host on-premises.

Before you begin

You should have the following:

- Root privileges to install the Console agent.

- Details about a proxy server, if a proxy is required for internet access from the Console agent.

  You have the option to configure a proxy server after installation but doing so requires restarting the Console agent.

- A CA-signed certificate, if the proxy server uses HTTPS or if the proxy is an intercepting proxy.

> You cannot set a certificate for a transparent proxy server when manually installing the Console agent. If you need to set a certificate for a transparent proxy server, you must use the Maintenance Console after installation. Learn more about the Agent Maintenance Console.

About this task

After installation, the Console agent automatically updates itself if a new version is available.

Steps

1. If the *http_proxy* or *https_proxy* system variables are set on the host, remove them:

   ```
   unset http_proxy
   unset https_proxy
   ```

   If you don't remove these system variables, the installation fails.

2. Download the Console agent software and then copy it to the Linux host. You can download it either from the NetApp Console or the NetApp Support site.
   - NetApp Console: Go to **Agents > Management> Deploy agent > On-prem > Manual install**.

     Choose download the agent installer files or a URL to the files.

   - NetApp Support Site (needed if you don't already have access to the Console) NetApp Support Site,
3. Assign permissions to run the script.

   ```
   chmod +x NetApp_Console_Agent_Cloud_<version>
   ```

   Where <version> is the version of the Console agent that you downloaded.

4. If installing in a Government Cloud environment, disable the configuration checks. Learn how to disable configuration checks for manual installations.
5. Run the installation script.

   ```
   ./NetApp_Console_Agent_Cloud_<version> --proxy <HTTP or HTTPS proxy
   server> --cacert <path and file name of a CA-signed certificate>
   ```

   You'll need to add proxy information if your network requires a proxy for internet access. You can add an explicit proxy during installation. The `--proxy` and `--cacert` parameters are optional and you won't be prompted to add them. If you have an explicit proxy server, you will need to enter the parameters as shown.

If you want to configure a transparent proxy, you can do so after you've installed. Learn about the agent maintenance console

+

Here is an example configuring an explicit proxy server with a CA-signed certificate:

+

```
./NetApp_Console_Agent_Cloud_v4.0.0--proxy
https://user:password@10.0.0.30:8080/ --cacert /tmp/cacert/certificate.cer
```

+
`--proxy` configures the Console agent to use an HTTP or HTTPS proxy server using one of the following formats:

+
* http://address:port
* http://user-name:password@address:port
* http://domain-name%92user-name:password@address:port
* https://address:port
* https://user-name:password@address:port
* https://domain-name%92user-name:password@address:port

+
Note the following:

+
The user can be a local user or domain user.
For a domain user, you must use the ASCII code for a \ as shown above.
The Console agent doesn't support user names or passwords that include the @ character.
If the password includes any of the following special characters, you must escape that special character by prepending it with a backslash: & or !

+
For example:

+
http://bxpproxyuser:netapp1\!@address:3128

1. If you used Podman, you'll need to adjust the aardvark-dns port.
   a. SSH to the Console agent virtual machine.
   b. Open podman */usr/share/containers/containers.conf* file and modify the chosen port for Aardvark DNS service. For example, change it to 54.

      ```
      vi /usr/share/containers/containers.conf
      ```

      For example:

```
# Port to use for dns forwarding daemon with netavark in rootful
bridge
# mode and dns enabled.
# Using an alternate port might be useful if other DNS services
should
# run on the machine.
#
dns_bind_port = 54
```

c. Reboot the Console agent virtual machine.

What's next?

You'll need to register the Console agent within the NetApp Console.

Register the Console agent with NetApp Console

Log into the Console and associate the Console agent with your organization. How you log in depends on the mode in which you are using Console. If you are using the Console in standard mode, you log in through the SaaS website. If you are using the Console in restricted mode, you log in locally from the Console agent host.

Steps

1. Open a web browser and enter the Console agent host URL:

   The Console host URL can be a localhost, a private IP address, or a public IP address, depending on the configuration of the host. For example, if the Console agent is in the public cloud without a public IP address, you must enter a private IP address from a host that has a connection to the Console agent host.

2. Sign up or log in.
3. After you log in, set up the Console:
   a. Specify the Console organization to associate with the Console agent.
   b. Enter a name for the system.
   c. Under **Are you running in a secured environment?** keep restricted mode disabled.

      Restricted mode isn't supported when the Console agent is installed on-premises.

   d. Select **Let's start**.

Provide cloud provider credentials to NetApp Console

After you install and set up the Console agent, add your cloud credentials so that the Console agent has the required permissions to perform actions in AWS or Azure.

AWS

Before you begin

If you just created these AWS credentials, they may take a few minutes to become available. Wait a few minutes before you add the credentials to the Console.

Steps

1. Select **Administration > Credentials**.
2. Select **Organization credentials**.
3. Select **Add Credentials** and follow the steps in the wizard.
    a. **Credentials Location**: Select *Amazon Web Services > Agent.
    b. **Define Credentials**: Enter an AWS access key and secret key.
    c. **Marketplace Subscription**: Associate a Marketplace subscription with these credentials by subscribing now or by selecting an existing subscription.
    d. **Review**: Confirm the details about the new credentials and select **Add**.

You can now go to the NetApp Console to start using the Console agent.

Azure

Before you begin

If you just created these Azure credentials, they may take a few minutes to become available. Wait a few minutes before you add the credentials the Console agent.

Steps

1. Select **Administration > Credentials**.
2. Select **Add Credentials** and follow the steps in the wizard.
    a. **Credentials Location**: Select **Microsoft Azure > Agent**.
    b. **Define Credentials**: Enter information about the Microsoft Entra service principal that grants the required permissions:
        - Application (client) ID
        - Directory (tenant) ID
        - Client Secret
    c. **Marketplace Subscription**: Associate a Marketplace subscription with these credentials by subscribing now or by selecting an existing subscription.
    d. **Review**: Confirm the details about the new credentials and select **Add**.

Result

The Console agent now has the permissions that it needs to perform actions in Azure on your behalf. You can now go to the NetApp Console to start using the Console agent.

Install a Console agent on-premises using VCenter

If you are a VMWare user, you can use an OVA to install a Console agent in your VCenter. THe OVA download or URL is available through the NetApp Console.

When you install a Console agent with your VCenter tools, you can use the VM web console to perform maintenance tasks. Learn more about the VM console for the agent.

Prepare to install the Console agent

Before installation, make sure your VM host meets the requirements and the Console agent can access the internet and targeted networks. To use NetApp data services or Cloud Volumes ONTAP, create cloud provider credentials for the Console agent to perform actions on your behalf.

Review Console agent host requirements

Make sure your host machine meets installation requirements before installing the Console agent.

- CPU: 8 cores or 8 vCPUs
- RAM: 32 GB
- Disk space: 165 GB (thick provisioned)
- vSphere 7.0 or higher
- ESXi host 7.03 or higher

Install the agent in a vCenter environment rather than directly on an ESXi host.

Set up network access for the Console agent

Work with your network administrator to ensure the Console agent has outbound access to the required endpoints and connections to targeted networks.

Connections to target networks

The Console agent requires a network connection to the location where you're planning to create and manage systems. For example, the network where you plan to create Cloud Volumes ONTAP systems or a storage system in your on-premises environment.

Outbound internet access

The network location where you deploy the Console agent must have an outbound internet connection to contact specific endpoints.

Endpoints contacted from computers when using the web-based NetApp Console

Computers that access the Console from a web browser must have the ability to contact several endpoints. You'll need to use the Console to set up the Console agent and for day-to-day use of the Console.

Prepare networking for the NetApp console.

Endpoints contacted from the Console agent

The Console agent requires outbound internet access to contact the following endpoints to manage resources and processes within your public cloud environment for day-to-day operations.

The endpoints listed below are all CNAME entries.

You can't manage resources in Google Cloud with an Console agent installed on your premises. To manage Google Cloud resources, install an agent in Google Cloud.

AWS

When the Console agent is installed on-premises, it needs network access to the following AWS endpoints in order to manage NetApp systems (such as Cloud Volumes ONTAP) deployed in AWS.

Endpoints contacted from the Console agent

The Console agent requires outbound internet access to contact the following endpoints to manage resources and processes within your public cloud environment for day-to-day operations.

The endpoints listed below are all CNAME entries.

Endpoints	Purpose
AWS services (amazonaws.com): • CloudFormation • Elastic Compute Cloud (EC2) • Identity and Access Management (IAM) • Key Management Service (KMS) • Security Token Service (STS) • Simple Storage Service (S3)	To manage AWS resources. The endpoint depends on your AWS region. Refer to AWS documentation for details
Amazon FsX for NetApp ONTAP: • api.workloads.netapp.com	The web-based console contacts this endpoint to interact with the Workload Factory APIs to manage and operate FSx for ONTAP based workloads.
https://mysupport.netapp.com	To obtain licensing information and to send AutoSupport messages to NetApp support.
https://signin.b2c.netapp.com	To update NetApp Support Site (NSS) credentials or to add new NSS credentials to the NetApp Console.
https://support.netapp.com	To obtain licensing information and to send AutoSupport messages to NetApp support as well as to receive software updates for Cloud Volumes ONTAP.
https://api.bluexp.netapp.com https://netapp-cloud-account.auth0.com https://netapp-cloud-account.us.auth0.com https://console.netapp.com https://components.console.bluexp.netapp.com https://cdn.auth0.com	To provide features and services within the NetApp Console.

Endpoints	Purpose
https://bluexpinfraprod.eastus2.data.azurecr.io https://bluexpinfraprod.azurecr.io	To obtain images for Console agent upgrades. • When you deploy a new agent, the validation check tests connectivity to current endpoints. If you use previous endpoints, the validation check fails. To avoid this failure, skip the validation check. Although the previous endpoints are still supported, NetApp recommends updating your firewall rules to the current endpoints as soon as possible. Learn how to update your endpoint list. • When you update to the current endpoints in your firewall, your existing agents will continue to work.

Azure

When the Console agent is installed on-premises, it needs network access to the following Azure endpoints in order to manage NetApp systems (such as Cloud Volumes ONTAP) deployed in Azure.

Endpoints	Purpose
https://management.azure.com https://login.microsoftonline.com https://blob.core.windows.net https://core.windows.net	To manage resources in Azure public regions.
https://management.chinacloudapi.cn https://login.chinacloudapi.cn https://blob.core.chinacloudapi.cn https://core.chinacloudapi.cn	To manage resources in Azure China regions.
https://mysupport.netapp.com	To obtain licensing information and to send AutoSupport messages to NetApp support.
https://signin.b2c.netapp.com	To update NetApp Support Site (NSS) credentials or to add new NSS credentials to the NetApp Console.
https://support.netapp.com	To obtain licensing information and to send AutoSupport messages to NetApp support as well as to receive software updates for Cloud Volumes ONTAP.

Endpoints	Purpose
https://api.bluexp.netapp.com https://netapp-cloud-account.auth0.com https://netapp-cloud-account.us.auth0.com https://console.netapp.com https://components.console.bluexp.netapp.com https://cdn.auth0.com	To provide features and services within the NetApp Console.
https://bluexpinfraprod.eastus2.data.azurecr.io https://bluexpinfraprod.azurecr.io	To obtain images for Console agent upgrades. • When you deploy a new agent, the validation check tests connectivity to current endpoints. If you use previous endpoints, the validation check fails. To avoid this failure, skip the validation check. Although the previous endpoints are still supported, NetApp recommends updating your firewall rules to the current endpoints as soon as possible. Learn how to update your endpoint list. • When you update to the current endpoints in your firewall, your existing agents will continue to work.

Proxy server

NetApp supports both explicit and transparent proxy configurations. If you are using a transparent proxy, you only need to provide the certificate for the proxy server. If you are using an explicit proxy, you'll also need the IP address and credentials.

- IP address
- Credentials
- HTTPS certificate

Ports

There's no incoming traffic to the Console agent, unless you initiate it or if it is used as a proxy to send AutoSupport messages from Cloud Volumes ONTAP to NetApp Support.

- HTTP (80) and HTTPS (443) provide access to the local UI, which you'll use in rare circumstances.
- SSH (22) is only needed if you need to connect to the host for troubleshooting.
- Inbound connections over port 3128 are required if you deploy Cloud Volumes ONTAP systems in a subnet where an outbound internet connection isn't available.

  If Cloud Volumes ONTAP systems don't have an outbound internet connection to send AutoSupport messages, the Console automatically configures those systems to use a proxy server that's included with the Console agent. The only requirement is to ensure that the Console agent's security group allows inbound connections over port 3128. You'll need to open this port after you deploy the Console agent.

Enable NTP

If you're planning to use NetApp Data Classification to scan your corporate data sources, you should enable a Network Time Protocol (NTP) service on both the Console agent and the NetApp Data Classification system so that the time is synchronized between the systems. Learn more about NetApp Data classification

Create Console agent cloud permissions for AWS or Azure

If you want to use NetApp data services in AWS or Azure with an on-premises Console agent, then you need to set up permissions in your cloud provider so that you can add the credentials to the Console agent after you install it.

You can't manage resources in Google Cloud with a Console agent installed on your premises. If you want to manage Google Cloud resources, you need to install an agent in Google Cloud.

AWS

For on-premises Console agents, provide AWS permissions by adding IAM user access keys.

Use IAM user access keys for on-premises Console agents; IAM roles are not supported for on-premises Console agents.

Steps

1. Log in to the AWS console and navigate to the IAM service.
2. Create a policy:
   a. Select **Policies > Create policy**.
   b. Select **JSON** and copy and paste the contents of the IAM policy for the Console agent.
   c. Finish the remaining steps to create the policy.

      Depending on the NetApp data services that you're planning to use, you might need to create a second policy.

      For standard regions, the permissions are spread across two policies. Two policies are required due to a maximum character size limit for managed policies in AWS. Learn more about IAM policies for the Console agent.
3. Attach the policies to an IAM user.
   - AWS Documentation: Creating IAM Roles
   - AWS Documentation: Adding and Removing IAM Policies
4. Ensure that the user has an access key that you can add to the NetApp Console after you install the Console agent.

Result

You should now have IAM user access keys with the required permissions.After you install the Console agent, associate these credentials with the Console agent from the Console.

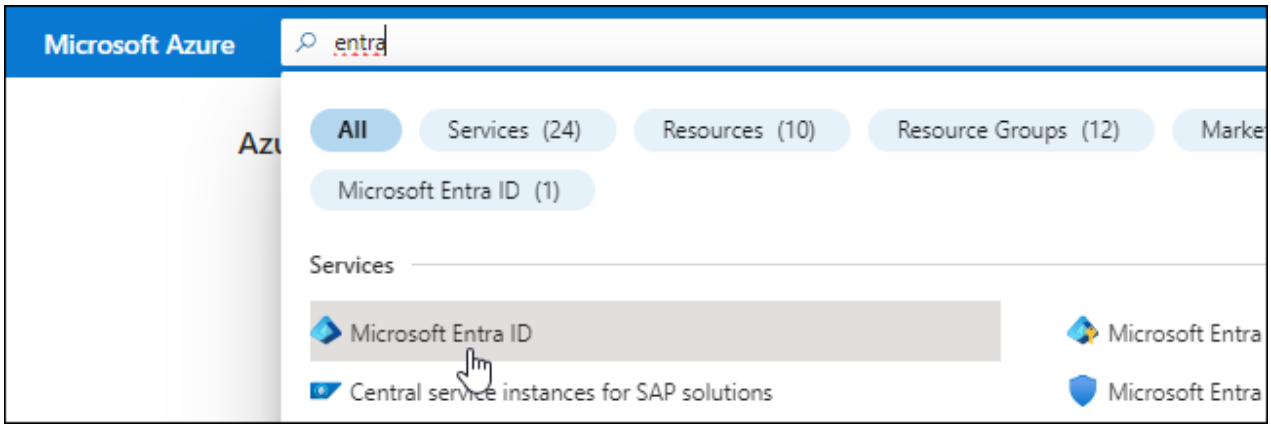
Azure

When the Console agent is installed on-premises, you need to give the Console agent Azure permissions by setting up a service principal in Microsoft Entra ID and getting the Azure credentials that the Console agent needs.

Create a Microsoft Entra application for role-based access control

1. Ensure that you have permissions in Azure to create an Active Directory application and to assign the application to a role.

   For details, refer to Microsoft Azure Documentation: Required permissions
2. From the Azure portal, open the **Microsoft Entra ID** service.



3. In the menu, select **App registrations**.
4. Select **New registration**.
5. Specify details about the application:
   - **Name**: Enter a name for the application.
   - **Account type**: Select an account type (any will work with the NetApp Console).
   - **Redirect URI**: You can leave this field blank.
6. Select **Register**.

   You've created the AD application and service principal.

Assign the application to a role

1. Create a custom role:

   Note that you can create an Azure custom role using the Azure portal, Azure PowerShell, Azure CLI, or REST API. The following steps show how to create the role using the Azure CLI. If you would prefer to use a different method, refer to Azure documentation

   a. Copy the contents of the custom role permissions for the Console agent and save them in a JSON file.

   b. Modify the JSON file by adding Azure subscription IDs to the assignable scope.

      You should add the ID for each Azure subscription from which users will create Cloud Volumes ONTAP systems.

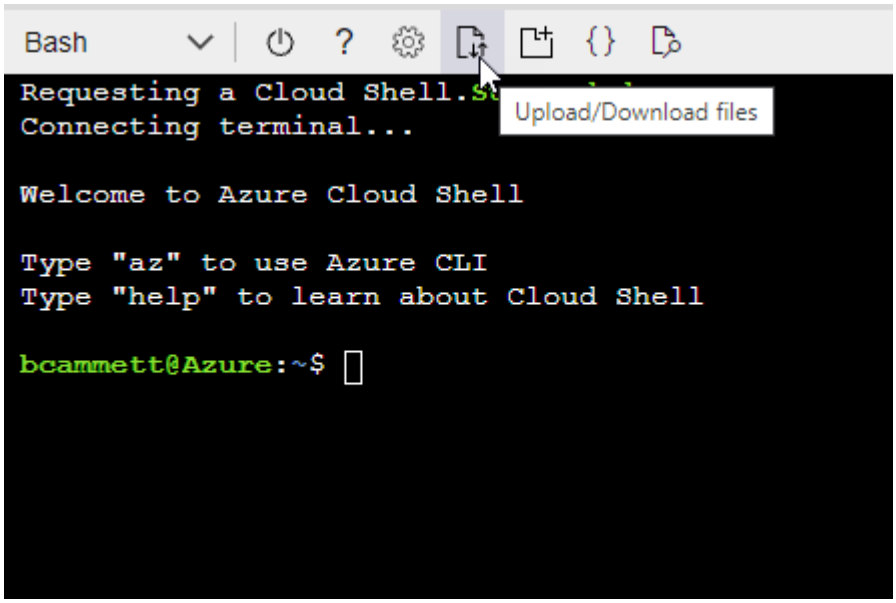
      Example

```
"AssignableScopes": [
"/subscriptions/d333af45-0d07-4154-943d-c25fbzzzzzzz",
"/subscriptions/54b91999-b3e6-4599-908e-416e0zzzzzzz",
"/subscriptions/398e471c-3b42-4ae7-9b59-ce5bbzzzzzzz"
]
```

   c. Use the JSON file to create a custom role in Azure.

      The following steps describe how to create the role by using Bash in Azure Cloud Shell.

- Start Azure Cloud Shell and choose the Bash environment.
- Upload the JSON file.

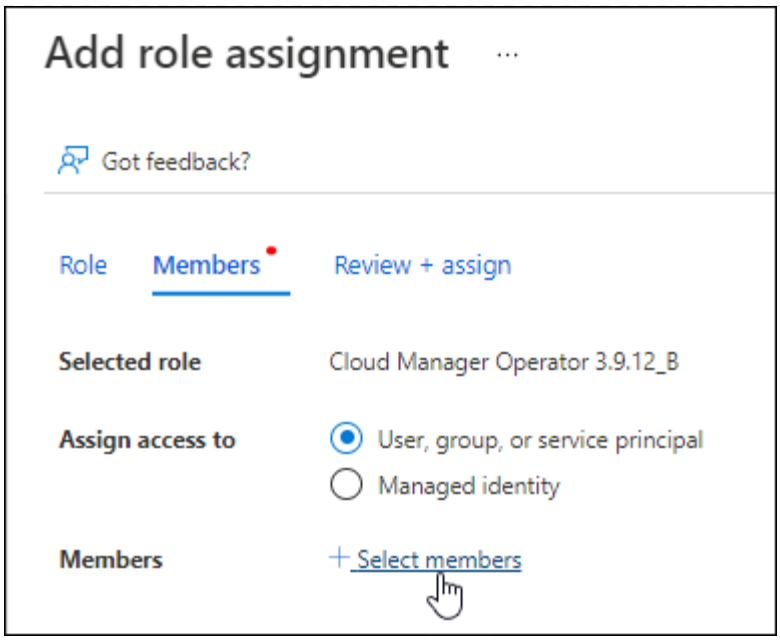


- Use the Azure CLI to create the custom role:

```
az role definition create --role-definition agent_Policy.json
```

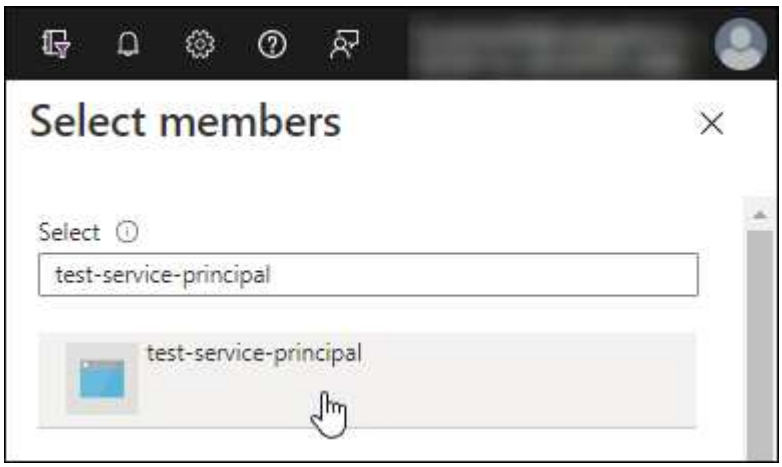
You should now have a custom role called Console Operator that you can assign to the Console agent virtual machine.

2. Assign the application to the role:
   a. From the Azure portal, open the **Subscriptions** service.
   b. Select the subscription.
   c. Select **Access control (IAM) > Add > Add role assignment**.
   d. In the **Role** tab, select the **Console Operator** role and select **Next**.
   e. In the **Members** tab, complete the following steps:
      - Keep **User, group, or service principal** selected.
      - Select **Select members**.



- Search for the name of the application.

  Here's an example:



- Select the application and select **Select**.
- Select **Next**.

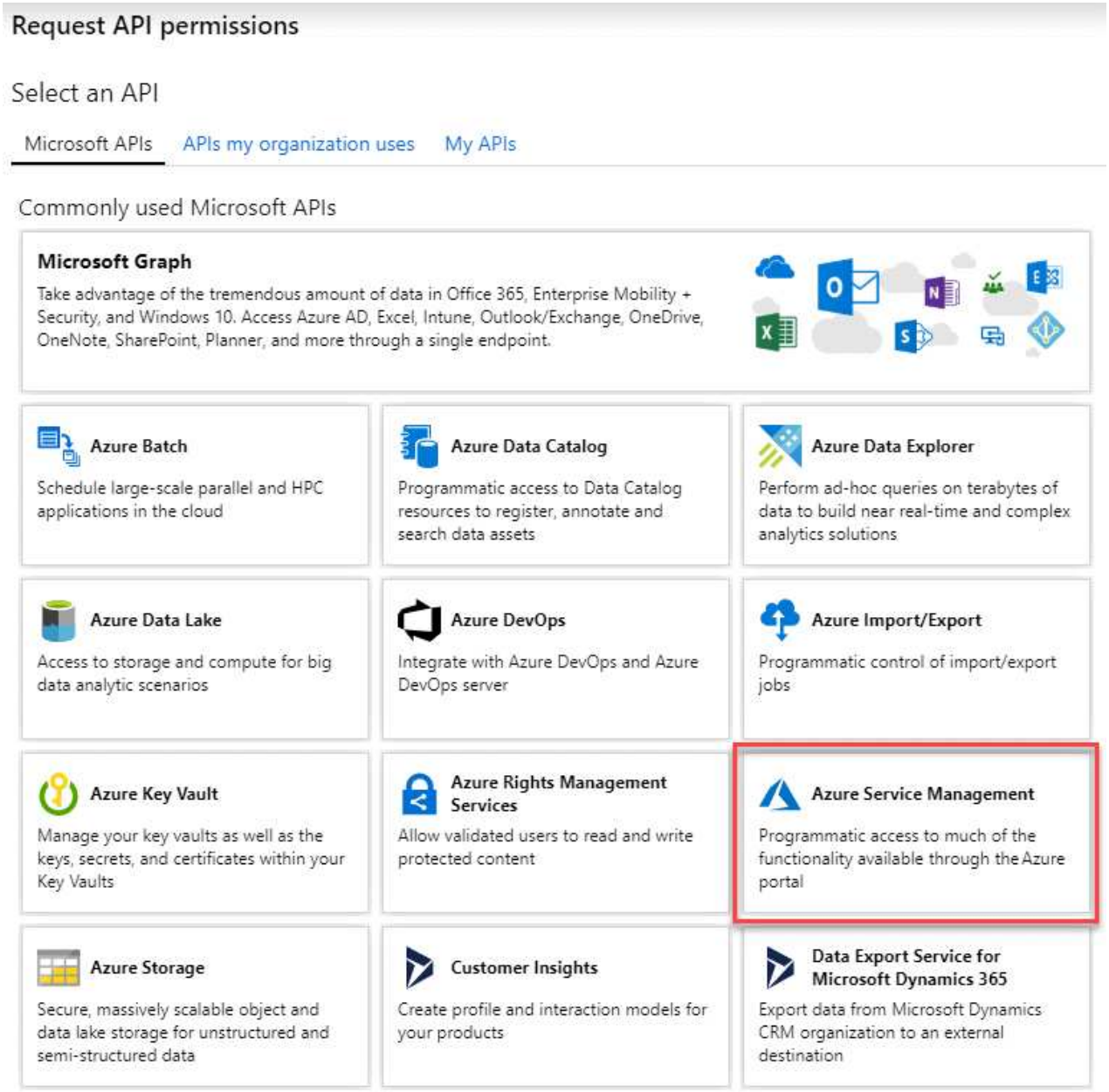
f. Select **Review + assign**.

The service principal now has the required Azure permissions to deploy the Console agent.

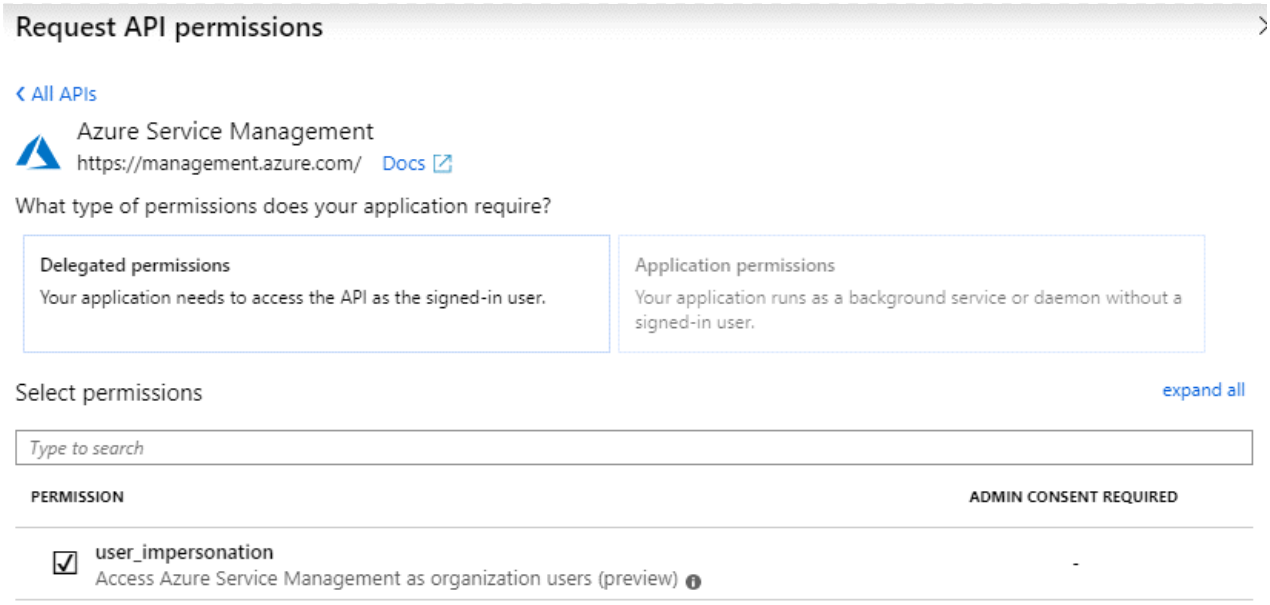
If you want to deploy Cloud Volumes ONTAP from multiple Azure subscriptions, then you must bind the service principal to each of those subscriptions. In the NetApp Console, you can select the subscription that you want to use when deploying Cloud Volumes ONTAP.

Add Windows Azure Service Management API permissions

1. In the **Microsoft Entra ID** service, select **App registrations** and select the application.
2. Select **API permissions > Add a permission**.
3. Under **Microsoft APIs**, select **Azure Service Management**.

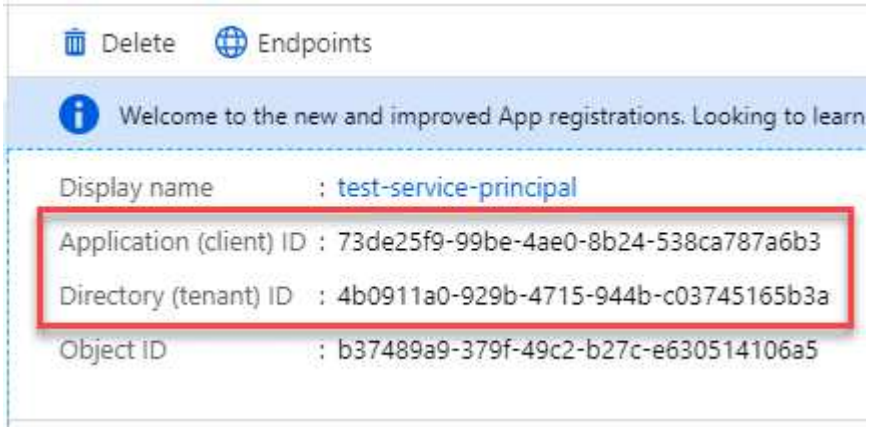


4. Select **Access Azure Service Management as organization users** and then select **Add permissions**.



Get the application ID and directory ID for the application

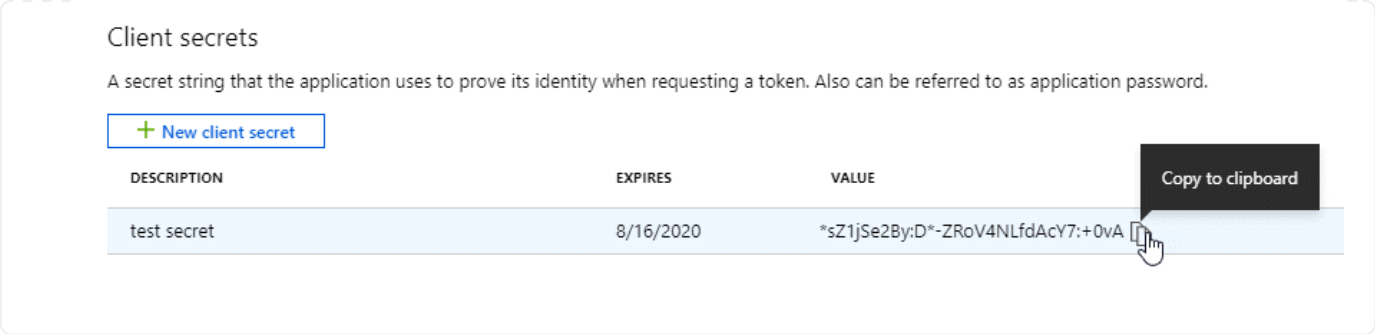
1. In the **Microsoft Entra ID** service, select **App registrations** and select the application.
2. Copy the **Application (client) ID** and the **Directory (tenant) ID**.



When you add the Azure account to the Console, you need to provide the application (client) ID and the directory (tenant) ID for the application. The Console uses the IDs to programmatically sign in.

Create a client secret

1. Open the **Microsoft Entra ID** service.
2. Select **App registrations** and select your application.
3. Select **Certificates & secrets > New client secret**.
4. Provide a description of the secret and a duration.
5. Select **Add**.
6. Copy the value of the client secret.



Install a Console agent in your VCenter environment

NetApp supports installing the Console agent in your VCenter environment. The OVA file includes a pre-configured VM image that you can deploy in your VMware environment. A file download or URL deployment is available directly from the NetApp Console. It includes the Console agent software and a self-signed certificate.

Download the OVA or copy the URL

Download the OVA or copy the OVA URL directly from the NetApp Console.

1. Select **Administration > Agents**.
2. On the **Overview** page, select **Deploy agent > On-Premises**.
3. Select **With OVA**.
4. Choose to either download the OVA or copy the URL to use in VCenter.

Deploy the agent in your VCenter

Log into your VCenter environment to deploy the agent.

Steps

1. Upload the self-signed certificate to your trusted certificates if your environment requires it. You replace this certificate after installation.Learn how to replace the self-signed certificate.
2. Deploy the OVA from the content library or local system.

From the local system	From the content library
a. Right-click and select **Deploy OVF template…**. b. Choose the OVA file from the URL or browse to its location, then select **Next**.	a. Go to your content library and select the Console agent OVA. b. Select **Actions** > **New VM from this template**

3. Complete the Deploy OVF Template wizard to deploy the Console agent.
4. Select a name and folder for the VM, then select **Next**.
5. Select a compute resource, then select **Next**.
6. Review the details of the template, then select **Next**.
7. Accept the license agreement, then select **Next**.
8. Choose the type of proxy configuration you want to use: explicit proxy, transparent proxy, or no proxy.

9. Select the datastore where you want to deploy the VM, then select **Next**. Be sure it meets host requirements.
10. Select the network to which you want to connect the VM, then select **Next**. Ensure the network is IPv4 and has outbound internet access to the required endpoints.
11. in the **Customize template** window, complete the following fields:
    - **Proxy information**
      - If you selected explicit proxy, enter the proxy server hostname or IP address and port number, as well as the username, password.
      - If you selected transparent proxy, upload the respective certificate.
    - **Virtual Machine Configuration**
      - **Skip config check**: This check box is unchecked by default which means the agent runs a configuration check to validate network access.
        - NetApp recommends leaving this box unchecked so that the installation includes a configuration check of the agent. The Configuration check validates that the agent has network access to the required endpoints. If it deployment fails because of connectivity issues, you can access the validation report and logs from the agent host. In some cases, if you are confident that the agent has network access, you can choose to skip the check. For example, if you are still using the previous endpoints used for agent upgrades, the validation fails with an error. To avoid this, mark the check box to install without a validation check. Learn how to update your endpoint list.
      - **Maintenance password**: Set the password for the `maint` user that allows access to the agent maintenance console.
      - **NTP servers**: Specify one or more NTP servers for time synchronization.
      - **Hostname**: Set the hostname for this VM. It must not include the search domain. For example, an FQDN of console10.searchdomain.company.com should be entered as console10.
      - **Primary DNS**: Specify the primary DNS server to use for name resolution.
      - **Secondary DNS**: Specify the secondary DNS server to use for name resolution.
      - Search domains: Specify the search domain name to use when resolving the hostname. For example, if the FQDN is console10.searchdomain.company.com, then enter searchdomain.company.com.
      - **IPv4 address**: The IP address that is mapped to the hostname.
      - **IPv4 subnet mask**: The subnet mask for the IPv4 address.
      - **IPv4 gateway address**: The gateway address for the IPv4 address.
12. Select **Next**.
13. Review the details in the **Ready to complete** window, select **Finish**.

    The vSphere task bar shows the progress as the Console agent is deployed.
14. Power on the VM.

If the deployment fails, you can access the validation report and logs from the agent host. Learn how to troubleshoot installation issues.

Register the Console agent with NetApp Console

Log into the Console and associate the Console agent with your organization. How you log in depends on the mode in which you are using Console. If you are using the Console in standard mode, you log in through the SaaS website. If you are using the Console in restricted or private mode, you log in locally from the Console agent host.

Steps

1. Open a web browser and enter the Console agent host URL:

   The Console host URL can be a localhost, a private IP address, or a public IP address, depending on the configuration of the host. For example, if the Console agent is in the public cloud without a public IP address, you must enter a private IP address from a host that has a connection to the Console agent host.

2. Sign up or log in.
3. After you log in, set up the Console:
   a. Specify the Console organization to associate with the Console agent.
   b. Enter a name for the system.
   c. Under **Are you running in a secured environment?** keep restricted mode disabled.

      Restricted mode isn't supported when the Console agent is installed on-premises.

   d. Select **Let's start**.

Add cloud provider credentials to the Console

After you install and set up the Console agent, add your cloud credentials so that the Console agent has the required permissions to perform actions in AWS or Azure.

AWS

Before you begin

If you just created these AWS credentials, they may take a few minutes to become available. Wait a few minutes before you add the credentials to the Console.

Steps

1. Select **Administration > Credentials**.
2. Select **Organization credentials**.
3. Select **Add Credentials** and follow the steps in the wizard.
    a. **Credentials Location**: Select *Amazon Web Services > Agent.
    b. **Define Credentials**: Enter an AWS access key and secret key.
    c. **Marketplace Subscription**: Associate a Marketplace subscription with these credentials by subscribing now or by selecting an existing subscription.
    d. **Review**: Confirm the details about the new credentials and select **Add**.

You can now go to the NetApp Console to start using the Console agent.

Azure

Before you begin

If you just created these Azure credentials, they may take a few minutes to become available. Wait a few minutes before you add the credentials the Console agent.

Steps

1. Select **Administration > Credentials**.
2. Select **Add Credentials** and follow the steps in the wizard.
    a. **Credentials Location**: Select **Microsoft Azure > Agent**.
    b. **Define Credentials**: Enter information about the Microsoft Entra service principal that grants the required permissions:
        - Application (client) ID
        - Directory (tenant) ID
        - Client Secret
    c. **Marketplace Subscription**: Associate a Marketplace subscription with these credentials by subscribing now or by selecting an existing subscription.
    d. **Review**: Confirm the details about the new credentials and select **Add**.

Result

THe Console agent now has the permissions that it needs to perform actions in Azure on your behalf. You can now go to the NetApp Console to start using the Console agent.

Ports for the on-premises Console agent

The Console agent uses *inbound* ports when installed manually on an on-premises Linux host. Refer to these ports for planning purposes.

These inbound rules apply to all NetApp Console deployment modes.

Protocol	Port	Purpose
HTTP	80	• Provides HTTP access from client web browsers to the local user interface • Used during the Cloud Volumes ONTAP upgrade process
HTTPS	443	Provides HTTPS access from client web browsers to the local user interface

Copyright information

Trademark information