Get started with NetApp Console (private mode)

NetApp Console setup and administration

NetApp
February 26, 2026

Table of Contents

Get started with NetApp Console (private mode) . . . 1
- Getting started workflow for NetApp Console in private mode . . . 1
  - Supported features and services for NetApp Console in private mode . . . 1
- Prepare for deployment in private mode . . . 4
  - Step 1: Understand how private mode works . . . 5
  - Step 2: Review installation options . . . 5
  - Step 3: Review host requirements . . . 5
  - Step 4: Install Podman or Docker Engine . . . 7
  - Step 5: Prepare networking . . . 10
  - Step 6: Prepare cloud permissions . . . 12
  - Step 7: Enable Google Cloud APIs . . . 22
- Deploy the Console agent in private mode . . . 23
  - Step 1: Install the Console agent . . . 23
  - Step 2: Set up NetApp Console . . . 24
  - Step 3: Provide permissions to Console agent . . . 24
- What you can do next (private mode) . . . 26

Get started with NetApp Console (private mode)

Getting started workflow for NetApp Console in private mode

You can install NetApp Console on-premises or in a secure cloud environment. Private mode is typically used with on-premises environments that have no internet connection and with secure cloud regions, which includes AWS Secret Cloud, AWS Top Secret Cloud, and Azure IL6

1 **Prepare for deployment**

a. Prepare a dedicated Linux host that meets requirements for CPU, RAM, disk space, container orchestration tool, and more.

b. Set up networking that provides access to the target networks.

c. For cloud deployments, set up permissions in your cloud provider so that you can associate those permissions with the Console agent after you install the software.

2 **Deploy the Console agent**

a. Install the Console agent on your own Linux host.

b. Set up NetApp Console by opening a web browser and entering the Linux host's IP address.

c. For cloud deployments, provide NetApp Console with the permissions that you previously set up.

Supported features and services for NetApp Console in private mode

The following table lists the storage management features, data services, and administrative features available in NetApp Console when you install it in private mode. Features differ when you install the NetApp Console in the cloud versus your premises.

Private mode installed in the cloud

Product area	NetApp service or feature	Console agent required
Storage systems management This portion of the table lists support for managing storage systems from the NetApp Console. It does not indicate the supported backup destinations for NetApp Backup and Recovery.	Cloud Volumes ONTAP	Yes, but because there's no internet access, the following features aren't available: automated software upgrades and AutoSupport.
	On-premises ONTAP clusters • Requires connectivity from the cloud (where the Console agent is installed) to the on-premises environment. You must have an installed Console agent to discover on-premises ONTAP clusters.	You can use direct discovery without a Console agent, but you need an agent to use management features.
Data Services	Backup and Recovery • In private mode, NetApp Backup and Recovery supports back up and restore of ONTAP volume data only. View the list of supported backup destinations for ONTAP data Back up and restore of application data and virtual machine data is not supported. • Not supported in Google Cloud or in AWS Secret Cloud, AWS Top Secret Cloud, or Azure IL6	Yes
	NetApp Classification • The only supported data sources are the ones that you can discover locally. View the sources that you can discover locally • Features that require outbound internet access are not supported. View the feature limitations	Yes
	NetApp Replication	Yes* (not needed if replicating from ONTAP on-premises to ONTAP on-premises)
NetApp support contract features	None	—

Product area	NetApp service or feature	Console agent required
Administrative features	Audit	No
	Licenses and subscriptions management • Only BYOL is supported with private mode. For Cloud Volumes ONTAP BYOL, only node-based licensing is supported. Capacity-based licensing is not supported. Because an outbound internet connection isn't available, you need to manually upload your Cloud Volumes ONTAP licensing file in the Console. Learn how to add licenses	No
	Identity and access management	No
	Manage cloud provider credentials	No
	Read-only mode	No
	Notifications	No

Private mode installed on-premises

Product area	NetApp service or feature	Console agent required
Storage systems management This portion of the table lists support for managing storage systems from the NetApp Console. It does not indicate the supported backup destinations for NetApp Backup and Recovery.	On-premises ONTAP clusters • Requires connectivity from the cloud (where the Console agent is installed) to the on-premises environment. You must have an installed Console agent to discover on-premises ONTAP clusters.	You can use direct discovery without a Console agent, but you need an agent to use management features.

Product area	NetApp service or feature	Console agent required
Data Services	Backup and Recovery • In private mode, NetApp Backup and Recovery supports back up and restore of ONTAP volume data only. View the list of supported backup destinations for ONTAP data • Back up and restore of application data and virtual machine data is not supported.	Yes
	NetApp Classification • The only supported data sources are the ones that you can discover locally. View the sources that you can discover locally • Features that require outbound internet access are not supported. View the feature limitations	Yes
	NetApp Replication	Yes* (not needed if replicating from ONTAP on-premises to ONTAP on-premises)
NetApp support contract features	None	N/A
Administrative features	Audit	No
	Licenses and subscriptions management • Only BYOL is supported with private mode.	No
	Identity and access management	No
	Manage cloud provider credentials	No
	Read-only mode	No
	Notifications	No

Prepare for deployment in private mode

Prepare your environment before you deploy NetApp Console in private mode. You need to review host requirements, prepare networking, set up permissions, and more.

To use NetApp Console in the AWS Secret Cloud or the AWS Top Secret Cloud, follow the specific instructions for those environments. Learn how to get started with Cloud Volumes ONTAP in the AWS Secret Cloud or Top Secret Cloud

Step 1: Understand how private mode works

In private mode, you install a Console agent either on-premises or in the cloud and then use NetApp Console to manage your storage systems (ONTAP on-premises and Cloud Volumes ONTAP). There is no connectivity to the NetApp Console API endpoints or the NetApp Console SaaS application, so you access the Console from the local UI provided by the Console agent.

Learn how private mode works.

Step 2: Review installation options

In private mode, you can install the Console agent on-premises or in the cloud by manually installing the agent on your own Linux host.

Where you install the Console agent determines which NetApp Console services and features are available when using private mode. The Console agent must be installed in the cloud if you want to deploy and manage Cloud Volumes ONTAP.

Step 3: Review host requirements

The host must meet specific operating system requirements, RAM requirements, port requirements, and so on to run the Console agent.

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.

Operating system and container requirements

The Console agent is supported with the following operating systems when using the Console in private 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	9.1 to 9.4 8.6 to 8.10	3.9.42 or later with the Console in private mode	Podman version 4.6.1 or 4.9.4 View Podman configuration requirements.	Supported in enforcing mode or permissive mode
Ubuntu	22.04 LTS	3.9.29 or later	Docker Engine 23.0.6 to 26.0.0 26.0.0 is supported with *new* agent 3.9.44 or later installations	Not supported

Notes:

1. The Console agent is supported on English-language versions of these operating systems.
2. For RHEL, 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.

Hypervisor

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

CPU

8 cores or 8 vCPUs

RAM

32 GB

AWS EC2 instance type

An instance type that meets CPU and RAM requirements. NetApp recommends t3.2xlarge.

Azure VM size

An instance type that meets CPU and RAM requirements. NetApp recommends Standard_D8s_v3.

Google Cloud machine type

An instance type that meets CPU and RAM requirements. NetApp recommends n2-standard-8.

The Console agent is supported in Google Cloud on a VM instance with an OS that supports Shielded VM features

Disk space in /opt

100 GiB of space must be available

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

Disk space in /var

20 GiB of space must be available

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

Step 4: Install Podman or Docker Engine

You need to prepare the host for the Console agent by installing 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
```

Step 5: Prepare networking

Set up networking for the Console agent to manage resources in your public cloud. Other than having a virtual network and subnet for the Console agent, ensure that the following requirements are met.

Connections to target networks

The Console agent must have a network connection to the location where you plan to manage storage. For example, the VPC or VNet where you plan to deploy Cloud Volumes ONTAP, or the data center where your

on-premises ONTAP clusters reside.

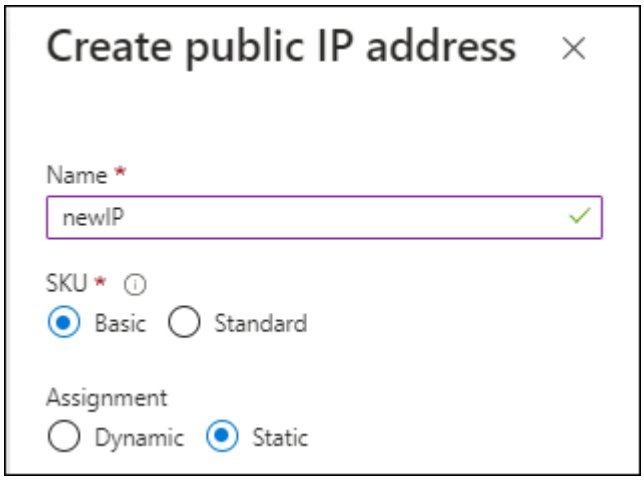
Endpoints for day-to-day operations

If you are planning to create Cloud Volumes ONTAP systems, the Console agent needs connectivity to endpoints in your cloud provider's publicly available resources.

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://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.azure.microsoft.scloud https://login.microsoftonline.microsoft.scloud https://blob.core.microsoft.scloud https://core.microsoft.scloud	To manage resources in the Azure IL6 region.
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://www.googleapis.com/compute/v1/ https://compute.googleapis.com/compute/v1 https://cloudresourcemanager.googleapis.com/v1/projects https://www.googleapis.com/compute/beta https://storage.googleapis.com/storage/v1 https://www.googleapis.com/storage/v1 https://iam.googleapis.com/v1 https://cloudkms.googleapis.com/v1 https://config.googleapis.com/v1/projects	To manage resources in Google Cloud.

Public IP address in Azure

If you want to use a public IP address with the Console agent VM in Azure, the IP address must use a Basic SKU to ensure that the Console uses this public IP address.



If you use a Standard SKU IP address instead, then the Console uses the *private* IP address of the Console agent, instead of the public IP. If the machine that you're using to access the Console doesn't have access to that private IP address, then actions from the Console will fail.

Azure documentation: Public IP SKU

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

  With private mode, the only time that NetApp Console sends outbound traffic is to your cloud provider in order to create a Cloud Volumes ONTAP system.

Ports

There's no incoming traffic to the Console agent, unless you initiate it.

HTTP (80) and HTTPS (443) provide access to the NetApp Console. SSH (22) is needed if you need to connect to the host for troubleshooting.

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

Step 6: Prepare cloud permissions

If the Console agent is installed in the cloud and you plan to create Cloud Volumes ONTAP systems, NetApp Console requires cloud provider permissions. You need to set up permissions in your cloud provider and then associate those permission with the Console agent instance after you install it.

To view the required steps, select the authentication option that you'd like to use for your cloud provider.

AWS IAM role

Use an IAM role to provide the Console agent with permissions. You'll need to manually attach the role to the EC2 instance for the Console agent.

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.
3. Create an IAM role:
   a. Select **Roles > Create role**.
   b. Select **AWS service > EC2**.
   c. Add permissions by attaching the policy that you just created.
   d. Finish the remaining steps to create the role.

Result

You now have an IAM role for the Console agent EC2 instance.

AWS access key

Set up permissions and an access key for an IAM user. Provide NetApp Console with the AWS access key after you install the Console agent and set up NetApp Console.

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 Console 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

The account now has the required permissions.

Azure role

Create an Azure custom role with the required permissions. Assign this role to the Console agent VM.

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

Steps

1. Enable a system-assigned managed identity on the VM where you plan to install the Console agent so that you can provide the required Azure permissions through a custom role.

   Microsoft Azure documentation: Configure managed identities for Azure resources on a VM using the Azure portal

2. Copy the contents of the custom role permissions for the Console agent and save them in a JSON file.
3. Modify the JSON file by adding Azure subscription IDs to the assignable scope.

   You should add the ID for each Azure subscription that you want to use with the NetApp Console.

   Example

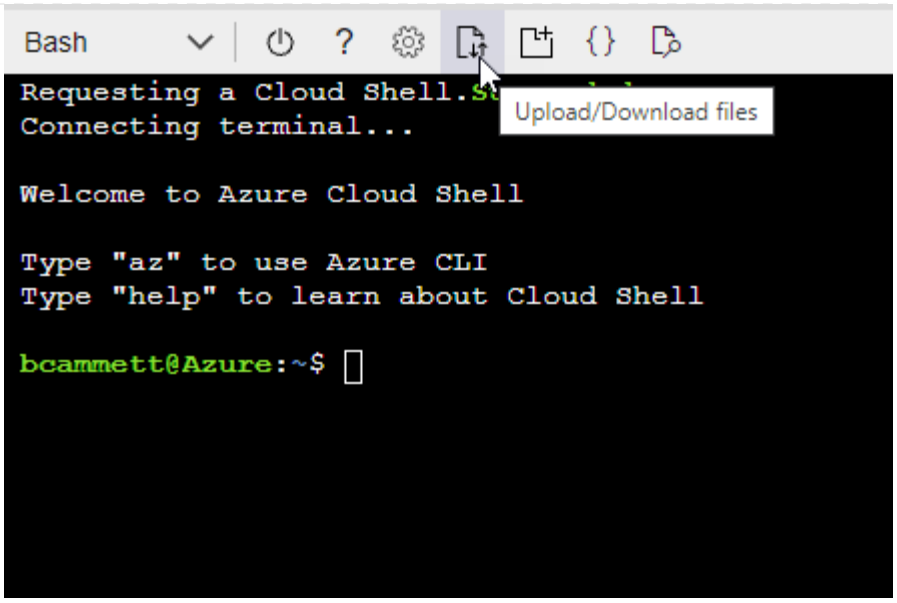
   ```
   "AssignableScopes": [
   "/subscriptions/d333af45-0d07-4154-943d-c25fbzzzzzzz",
   "/subscriptions/54b91999-b3e6-4599-908e-416e0zzzzzzz",
   "/subscriptions/398e471c-3b42-4ae7-9b59-ce5bbzzzzzzz"
   ]
   ```

4. 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.

   a. Start Azure Cloud Shell and choose the Bash environment.

   b. Upload the JSON file.



c. Use the Azure CLI to create the custom role:

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

Azure service principal

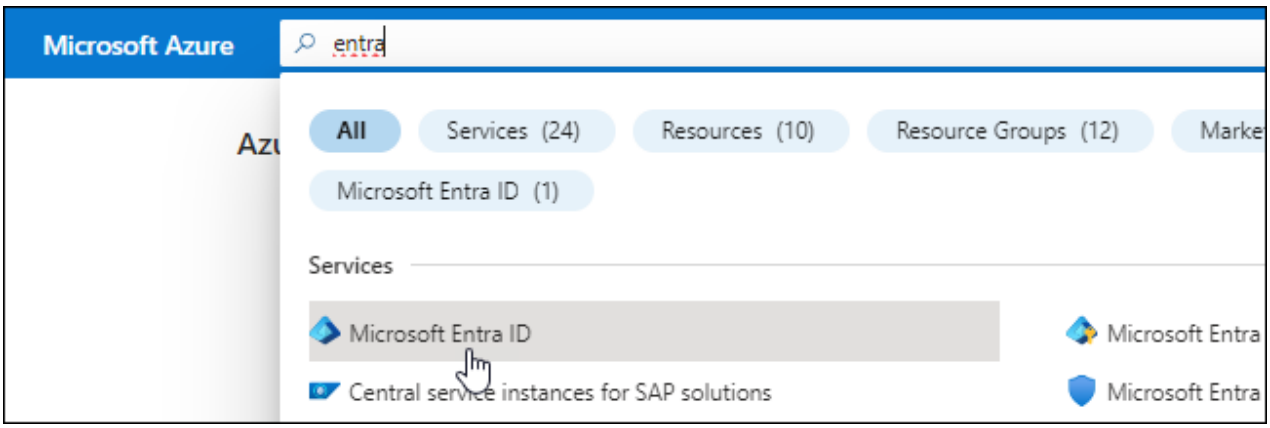
Create and set up a service principal in Microsoft Entra ID and obtain the Azure credentials that NetApp Console needs. You'll need to provide NetApp Console with these credentials after you install the Console agent and set up NetApp Console.

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.

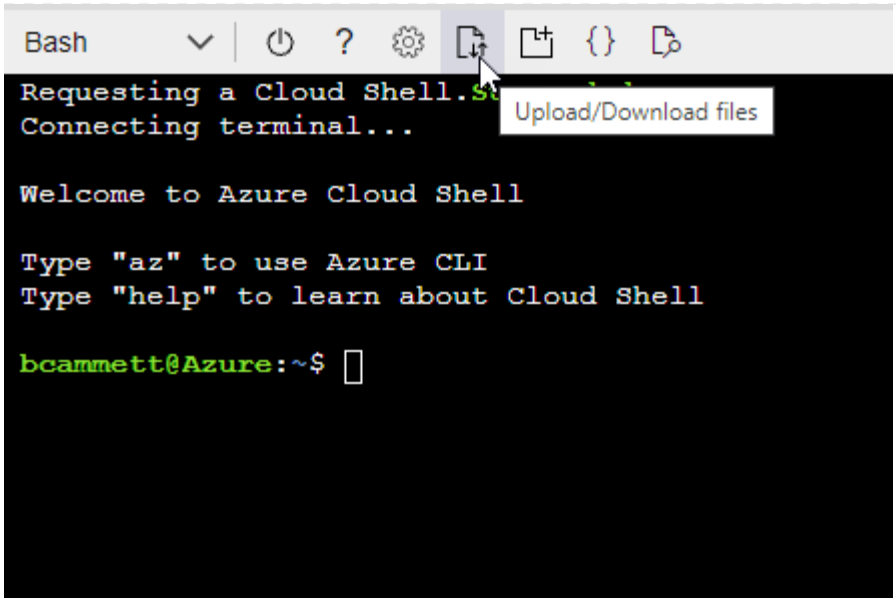
      Example

      ```
      "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.

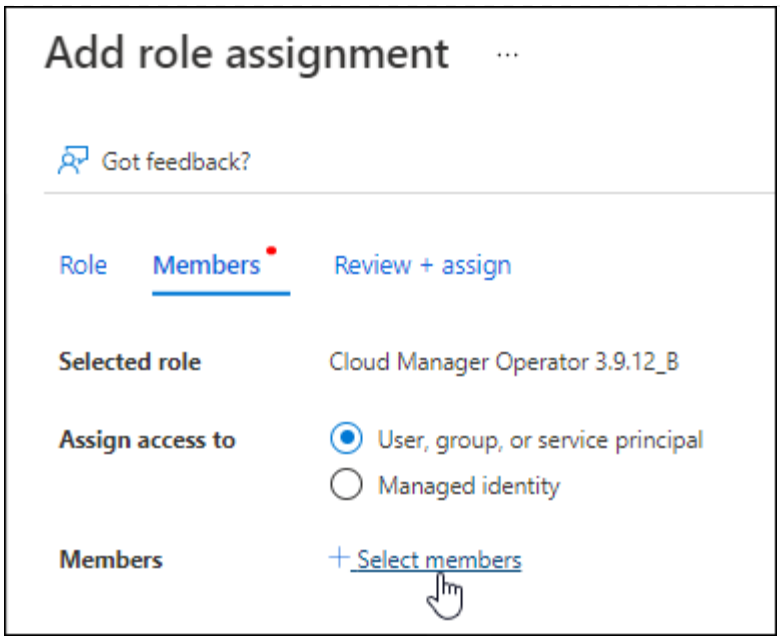


- Use the Azure CLI to create the custom role:

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

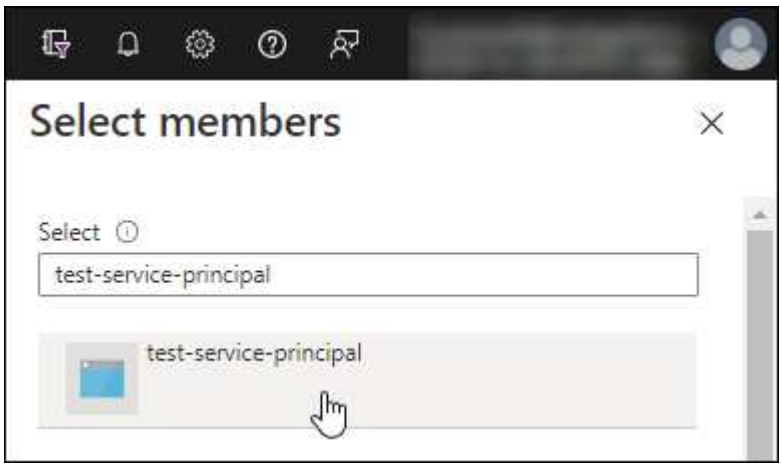
You should now have a custom role called Console Operator that you can assign to the Console agent virtual machine.

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**.

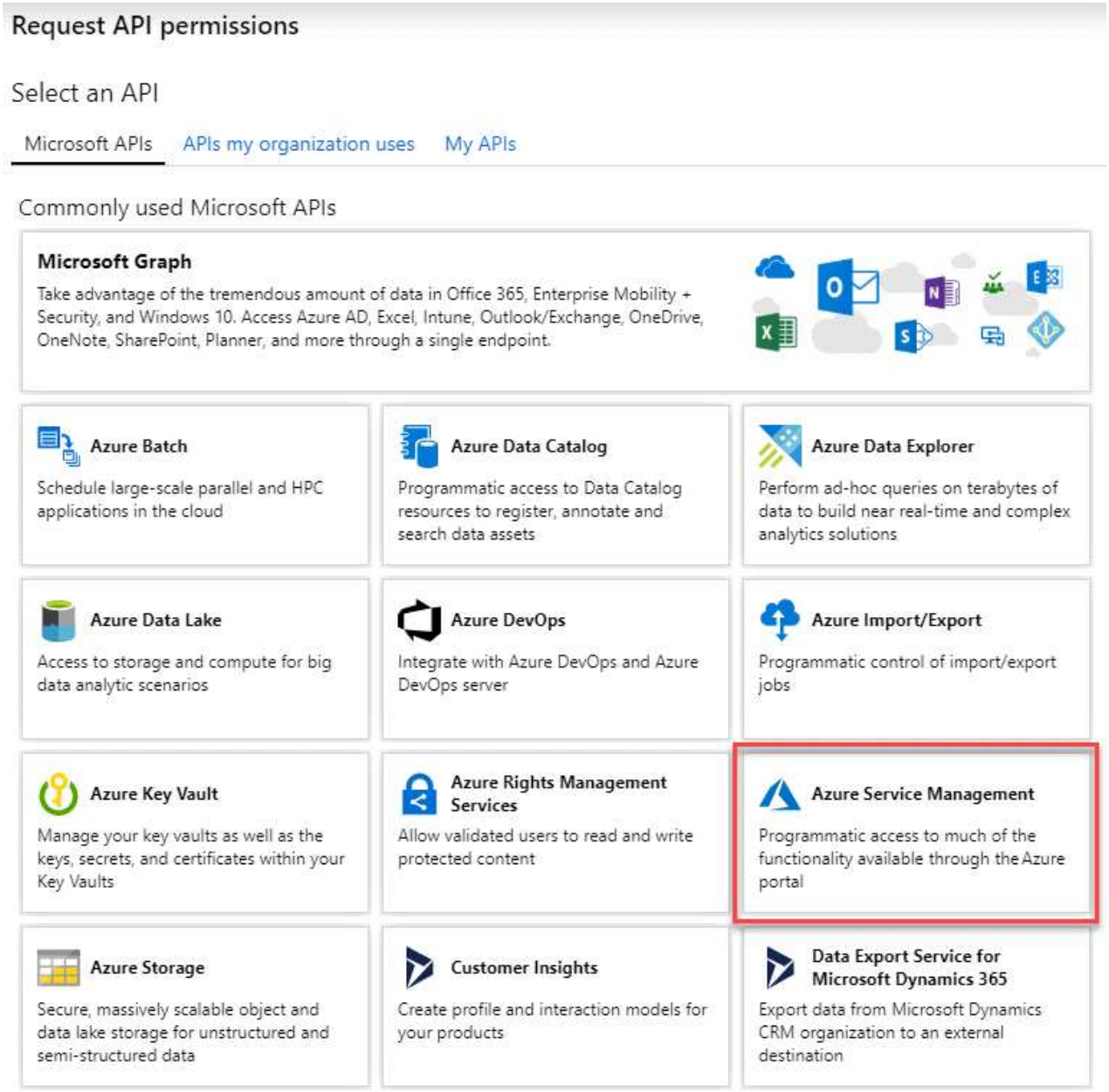
f. Select **Review + assign**.

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

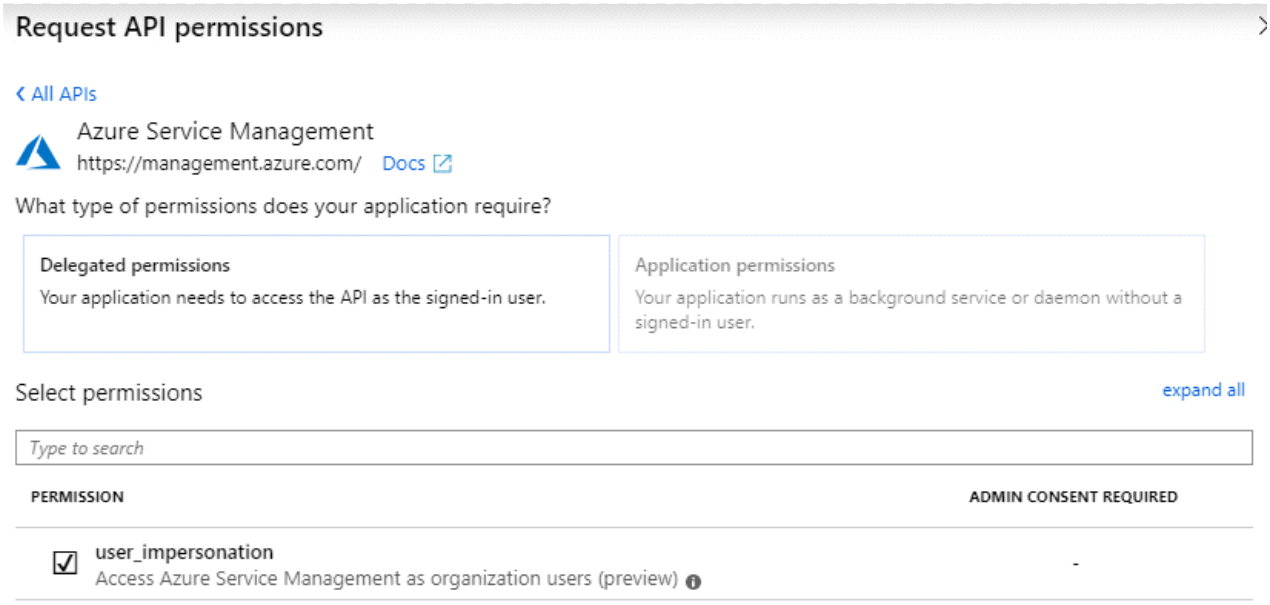
If you want to deploy Cloud Volumes ONTAP from multiple Azure subscriptions, then you must bind the service principal to each of those subscriptions. In the NetApp Console, you can select the subscription that you want to use when deploying Cloud Volumes ONTAP.

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**.

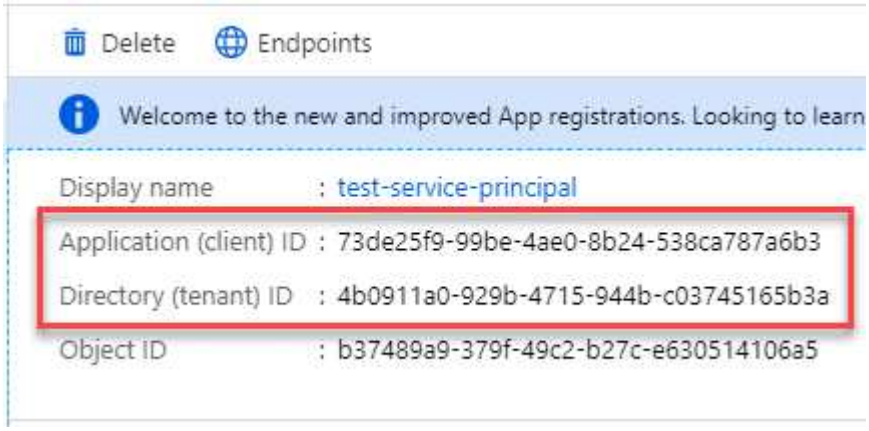


4. Select **Access Azure Service Management as organization users** and then select **Add permissions**.



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.

Result

Your service principal is now setup and you should have copied the application (client) ID, the directory (tenant) ID, and the value of the client secret. Enter this information in NetApp Console when you add an Azure credential.

Google Cloud service account

Create a role and apply it to a service account that you'll use for the Console agent VM instance.

Steps

1. Create a custom role in Google Cloud:
   a. Create a YAML file that includes the permissions defined in the Console agent policy for Google Cloud.
   b. From Google Cloud, activate cloud shell.
   c. Upload the YAML file that includes the required permissions for the Console agent.
   d. Create a custom role by using the `gcloud iam roles create` command.

      The following example creates a role named "agent" at the project level:

      ```
      gcloud iam roles create agent --project=myproject
      --file=agent.yaml
      ```

      Google Cloud docs: Creating and managing custom roles

2. Create a service account in Google Cloud:
   a. From the IAM & Admin service, select **Service Accounts > Create Service Account**.
   b. Enter service account details and select **Create and Continue**.
   c. Select the role that you just created.
   d. Finish the remaining steps to create the role.

      Google Cloud docs: Creating a service account

Step 7: Enable Google Cloud APIs

You need to enable several APIs to deploy Cloud Volumes ONTAP in Google Cloud.

Step

1. Enable the following Google Cloud APIs in your project
   - Cloud Build API (required for private mode Cloud Volumes ONTAP deployments using Infrastructure Manager)
   - Cloud Deployment Manager V2 API
   - Cloud Infrastructure Manager API
   - Cloud Logging API
   - Cloud Resource Manager API
   - Compute Engine API
   - Identity and Access Management (IAM) API
   - Cloud Key Management Service (KMS) API (Required only if you are planning to use NetApp Backup and Recovery with customer-managed encryption keys (CMEK))
   - Cloud Quotas API (required for Cloud Volumes ONTAP deployments using Infrastructure Manager)

Deploy the Console agent in private mode

Deploy the NetApp Console in private mode so that you can use it with no outbound connectivity. To get started, install a Console agent, set up the NetApp Console by accessing the user interface that's running on the Console agent, and then provide the cloud permissions that you previously set up.

Step 1: Install the Console agent

Download the product installer from the NetApp Support Site and then manually install the Console on your own Linux host.

To use the NetApp Console in the AWS Secret Cloud or the AWS Top Secret Cloud, then you should follow separate instructions to get started in those environments. Learn how to get started with Cloud Volumes ONTAP in the AWS Secret Cloud or Top Secret Cloud

Before you begin

- Root privileges are required to install the Console agent.
- Depending on your operating system, either Podman or Docker Engine is required before you install the Console agent.

Steps

1. Download the Console agent software from the NetApp Support Site

   Be sure to download the offline installer for private networks without internet access.

2. Copy the installer to the Linux host.
3. Assign permissions to run the script.

   ```
   chmod +x /path/NetApp-Console-Private-Mode-<version>
   ```

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

4. Run the installation script:

```
sudo /path/NetApp-Console-Private-Mode-<version>
```

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

Result

The Console agent is installed. You can now set up the NetApp Console.

Step 2: Set up NetApp Console

When you access the NetApp Console for the first time, you'll be prompted to set up NetApp Console.

Steps

1. Open a web browser and enter the IP address of the Linux host where you installed the Console agent.
2. Select **Set Up New Console agent** and follow the prompts to set up the system.
   - **System Details**: Enter a name for the Console agent and your company name.
   - **Create an Admin User**: Create the admin user for the system.
   - **Review**: Review the details, accept the license agreement, and then select **Set Up**.
3. Log in to NetApp Console using the admin user that you just created.

Result

The Console agent is now installed. Access its IP address in a web browser to access the NetApp Console in private mode.

When new versions of the Console software are available, they'll be posted to the NetApp Support Site. Learn how to upgrade the Console agent.

What's next?

Provide the Console agent with the cloud provider permissions that you previously set up.

Step 3: Provide permissions to Console agent

If you want to create Cloud Volumes ONTAP working environments, you'll need to provide the Console agent with the cloud provider permissions that you previously set up.

Learn how to prepare cloud permissions.

AWS IAM role

Attach the IAM role that you previously created to the Console agent EC2 instance.

Steps

1. Go to the Amazon EC2 console.
2. Select **Instances**.
3. Select the Console agent instance.
4. Select **Actions > Security > Modify IAM role**.
5. Select the IAM role and select **Update IAM role**.

AWS access key

Provide the Console agent with the AWS access key for an IAM user that has the required permissions.

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**.

Azure role

Go to the Azure portal and assign the Azure custom role to the Console agent virtual machine for one or more subscriptions.

Steps

1. From the Azure Portal, open the **Subscriptions** service and select your subscription.

   It's important to assign the role from the **Subscriptions** service because this specifies the scope of the role assignment at the subscription level. The *scope* defines the set of resources that the access applies to. If you specify a scope at a different level (for example, at the virtual machine level), your ability to complete actions from within the NetApp Console will be affected.

   Microsoft Azure documentation: Understand scope for Azure RBAC

2. Select **Access control (IAM)** > **Add** > **Add role assignment**.
3. In the **Role** tab, select the **Console Operator** role and select **Next**.

   Console Operator is the default name provided in the policy. If you chose a different name for the role, then select that name instead.

4. In the **Members** tab, complete the following steps:
   a. Assign access to a **Managed identity**.
   b. Select **Select members**, select the subscription in which the Console agent virtual machine was

created, under **Managed identity**, choose **Virtual machine**, and then select the Console agent virtual machine.

c. Select **Select**.

d. Select **Next**.

e. Select **Review + assign**.

f. If you want to manage resources in additional Azure subscriptions, switch to that subscription and then repeat these steps.

Azure service principal

Provide the Console agent with the credentials for the Azure service principal that you previously setup.

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**.

Google Cloud service account

Associate the service account with the Console agent VM.

Steps

1. Go to the Google Cloud portal and assign the service account to the Console agent VM instance.

   Google Cloud documentation: Changing the service account and access scopes for an instance

2. If you want to manage resources in other projects, grant access by adding the service account with the Console agent role to that project. You'll need to repeat this step for each project.

What you can do next (private mode)

After you get up and running with NetApp Console in private mode, you can start using the services that are supported with private mode.

For help, refer to the following documentation:

- Discover on-premises ONTAP clusters
- Scan on-premises ONTAP volume data using NetApp Classification
- Monitor license usage

Copyright information

Trademark information