Deploy a Console agent

NetApp Console setup and administration

NetApp
January 27, 2026

Table of Contents

Deploy a Console agent

AWS

Console agent installation options in AWS

There are a few different ways to create a Console agent in AWS. Directly from the NetApp Console is the most common way.

The following installation options are available:

- Create the Console agent directly from the Console (this is the standard option)

  This action launches an EC2 instance running Linux and the Console agent software in a VPC of your choice.

- Create a Console agent from the AWS Marketplace

  This action also launches an EC2 instance running Linux and the Console agent software, but the deployment is initiated directly from the AWS Marketplace, rather than from the Console.

- Download and manually install the software on your own Linux host

The installation option that you choose impacts how you prepare for installation. This includes how you provide the Console with the required permissions that it needs to authenticate and manage resources in AWS.

Create a Console agent in AWS from NetApp Console

You can create a Console agent in AWS directly from the NetApp Console. Before creating a Console agent in AWS from the Console, you need to set up your networking and prepare AWS permissions.

Before you begin

- You should have an understanding of Console agents.
- You should review Console agent limitations.

Step 1: Set up networking for deploying a Console agent in AWS

Ensure that the network location where you plan to install the Console agent supports the following requirements. These requirements enable the Console agent to manage resources and processes in your hybrid cloud.

VPC and subnet

When you create the Console agent, you need to specify the VPC and subnet where it should reside.

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 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.

Endpoints contacted from the NetApp console

As you use the web-based NetApp Console that's provided through the SaaS layer, it contacts several endpoints to complete data management tasks. This includes endpoints that are contacted to deploy the Console agent from the the Console.

View the list of endpoints contacted from the NetApp console.

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

You'll need to implement this networking requirement after you create the Console agent.

Step 2: Set up AWS permissions for the Console agent

The Console needs to authenticate with AWS before it can deploy the Console agent in your VPC. You can choose one of these authentication methods:

- Let the Console assume an IAM role that has the required permissions
- Provide an AWS access key and secret key for an IAM user who has the required permissions

With either option, the first step is to create an IAM policy. This policy contains only the permissions needed to launch the Console agent in AWS from the Console.

If needed, you can restrict the IAM policy by using the IAM `Condition` element. AWS documentation: Condition element

Steps

1. Go to the AWS IAM console.
2. Select **Policies > Create policy**.
3. Select **JSON**.
4. Copy and paste the following policy:

   This policy contains only the permissions needed to launch the Console agent in AWS from the Console. When the Console creates the Console agent, it applies a new set of permissions to the Console agent that enables the Console agent to manage AWS resources. View permissions required for the Console agent itself.

```
{
  "Version": "2012-10-17",
  "Statement": [
    {
      "Effect": "Allow",
      "Action": [
        "iam:CreateRole",
        "iam:DeleteRole",
        "iam:PutRolePolicy",
        "iam:CreateInstanceProfile",
        "iam:DeleteRolePolicy",
        "iam:AddRoleToInstanceProfile",
        "iam:RemoveRoleFromInstanceProfile",
        "iam:DeleteInstanceProfile",
        "iam:PassRole",
        "iam:ListRoles",
        "ec2:DescribeInstanceStatus",
```

```
        "ec2:RunInstances",
        "ec2:ModifyInstanceAttribute",
        "ec2:CreateSecurityGroup",
        "ec2:DeleteSecurityGroup",
        "ec2:DescribeSecurityGroups",
        "ec2:RevokeSecurityGroupEgress",
        "ec2:AuthorizeSecurityGroupEgress",
        "ec2:AuthorizeSecurityGroupIngress",
        "ec2:RevokeSecurityGroupIngress",
        "ec2:CreateNetworkInterface",
        "ec2:DescribeNetworkInterfaces",
        "ec2:DeleteNetworkInterface",
        "ec2:ModifyNetworkInterfaceAttribute",
        "ec2:DescribeSubnets",
        "ec2:DescribeVpcs",
        "ec2:DescribeDhcpOptions",
        "ec2:DescribeKeyPairs",
        "ec2:DescribeRegions",
        "ec2:DescribeInstances",
        "ec2:CreateTags",
        "ec2:DescribeImages",
        "ec2:DescribeAvailabilityZones",
        "ec2:DescribeLaunchTemplates",
        "ec2:CreateLaunchTemplate",
        "cloudformation:CreateStack",
        "cloudformation:DeleteStack",
        "cloudformation:DescribeStacks",
        "cloudformation:DescribeStackEvents",
        "cloudformation:ValidateTemplate",
        "ec2:AssociateIamInstanceProfile",
        "ec2:DescribeIamInstanceProfileAssociations",
        "ec2:DisassociateIamInstanceProfile",
        "iam:GetRole",
        "iam:TagRole",
        "kms:ListAliases",
        "cloudformation:ListStacks"
      ],
      "Resource": "*"
    },
    {
      "Effect": "Allow",
      "Action": [
        "ec2:TerminateInstances"
      ],
      "Condition": {
        "StringLike": {
```

```
            "ec2:ResourceTag/OCCMInstance": "*"
          }
        },
        "Resource": [
          "arn:aws:ec2:*:*:instance/*"
        ]
      }
    ]
}
```

5. Select **Next** and add tags, if needed.
6. Select **Next** and enter a name and description.
7. Select **Create policy**.
8. Either attach the policy to an IAM role that the Console can assume or to an IAM user so that you can provide the Console with access keys:
   - (Option 1) Set up an IAM role that the Console can assume:
     a. Go to the AWS IAM console in the target account.
     b. Under Access Management, select **Roles > Create Role** and follow the steps to create the role.
     c. Under **Trusted entity type**, select **AWS account**.
     d. Select **Another AWS account** and enter the ID of the Console SaaS account: 952013314444
     e. Select the policy that you created in the previous section.
     f. After you create the role, copy the Role ARN so that you can paste it in the Console when you create the Console agent.
   - (Option 2) Set up permissions for an IAM user so that you can provide the Console with access keys:
     a. From the AWS IAM console, select **Users** and then select the user name.
     b. Select **Add permissions > Attach existing policies directly**.
     c. Select the policy that you created.
     d. Select **Next** and then select **Add permissions**.
     e. Ensure that you have the access key and secret key for the IAM user.

Result

You should now have an IAM role that has the required permissions or an IAM user that has the required permissions. When you create the Console agent from the Console, you can provide information about the role or access keys.

Step 3: Create the Console agent

Create the Console agent directly from the the Console web-based console.

About this task

- Creating the Console agent from the Console deploys an EC2 instance in AWS using a default configuration. Do not switch to a smaller EC2 instance with fewer CPUs or less RAM after creating the Console agent. Learn about the default configuration for the Console agent.

- When the Console creates the Console agent, it creates an IAM role and a profile for the agent. This role includes permissions that enables the Console agent to manage AWS resources. Ensure the role is updated as new permissions are added in future releases.
  Learn more about the IAM policy for the Console agent.

Before you begin

You should have the following:

- An AWS authentication method: either an IAM role or access keys for an IAM user with the required permissions.
- A VPC and subnet that meets networking requirements.
- A key pair for the EC2 instance.
- Details about a proxy server, if a proxy is required for internet access from the Console agent.
- Set up networking requirements.
- Set up AWS permissions.

Steps

1. Select **Administration > Agents**.
2. On the **Overview** page, select **Deploy agent > AWS**
3. Follow the steps in the wizard to create the Console agent:
4. On the **Introduction** page provides an overview of the process
5. On the **AWS Credentials** page, specify your AWS region and then choose an authentication method, which is either an IAM role that the Console can assume or an AWS access key and secret key.

   If you choose **Assume Role**, you can create the first set of credentials from the Console agent deployment wizard. Any additional set of credentials must be created from the Credentials page. They will then be available from the wizard in a drop-down list. Learn how to add additional credentials.

6. On the **Details** page, provide details about the Console agent.
   - Enter a name.
   - Add custom tags (metadata).
   - Choose whether you want the Console to create a new role that has the required permissions, or if you want to select an existing role that you set up with the required permissions.
   - Choose whether you want to encrypt the Console agent's EBS disks. You have the option to use the default encryption key or to use a custom key.
7. On the **Network** page, Specify a VPC, subnet, and key pair for the agent, choose whether to enable a public IP address, and optionally specify a proxy configuration.

   Ensure you have the correct key pair to access the Console agent virtual machine. Without a key pair, you cannot access it.

8. On the **Security Group** page, choose whether to create a new security group or whether to select an existing security group that allows the required inbound and outbound rules.

   View security group rules for AWS.

9. Review your selections to verify that your set up is correct.

    a. The **Validate agent configuration** check box is marked by default to have the Console validate the network connectivity requirements when you deploy. If the Console fails to deploy the agent, it provides a report to help you troubleshoot. If the deployment succeeds, no report is provided.

    If you are still using the previous endpoints used for agent upgrades, the validation fails with an error. To avoid this, unmark the check box to skip the validation check.

10. Select **Add**.

    The Console deploys the agent in about 10 minutes. Stay on the page until the process completes.

Result

After the process is complete, the Console agent is available for use from the Console.

If the deployment fails, you can download a report and logs from the Console to help you fix the issues. Learn how to troubleshoot installation issues.

If you have Amazon S3 buckets in the same AWS account where you created the Console agent, you'll see an Amazon S3 working environment appear on the **Systems** page automatically. Learn how to manage S3 buckets from NetApp Console

Create a Console agent from the AWS Marketplace

You create a Console agent in AWS directly from the AWS Marketplace. To create a Console agent from the AWS Marketplace, you need to set up your networking, prepare AWS permissions, review instance requirements, and then create the Console agent.

Before you begin

- You should have an understanding of Console agents.
- You should review Console agent limitations.

Step 1: Set up networking

Ensure the network location for the Console agent meets the following requirements to manage hybrid cloud resources.

VPC and subnet

When you create the Console agent, you need to specify the VPC and subnet where it should reside.

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 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.

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

Implement this network access after you create the Console agent.

Step 2: Set up AWS permissions

To prepare for a marketplace deployment, create IAM policies in AWS and attach them to an IAM role. When you create the Console agent from the AWS Marketplace, you are prompted to select that 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.

      You may need to create a second policy based on the NetApp data services you plan to use. 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. 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 that you can associate with the EC2 instance during deployment from the AWS Marketplace.

Step 3: Review instance requirements

When you create the Console agent, you need to choose an EC2 instance type that meets the following requirements.

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.

Step 4: Create the Console agent

Create the Console agent directly from the AWS Marketplace.

About this task

Creating the Console agent from the AWS Marketplace deploys an EC2 instance in AWS using a default configuration. Learn about the default configuration for the Console agent.

Before you begin

You should have the following:

- A VPC and subnet that meets networking requirements.
- An IAM role with an attached policy that includes the required permissions for the Console agent.
- Permissions to subscribe and unsubscribe from the AWS Marketplace for your IAM user.
- An understanding of CPU and RAM requirements for the instance.
- A key pair for the EC2 instance.

Steps

1. Go to the NetApp Console agent listing on the AWS Marketplace
2. On the Marketplace page, select **Continue to Subscribe**.
3. To subscribe to the software, select **Accept Terms**.

   The subscription process can take a few minutes.

4. After the subscription process is complete, select **Continue to Configuration**.
5. On the **Configure this software** page, ensure that you've selected the correct region and then select **Continue to Launch**.
6. On the **Launch this software** page, under **Choose Action**, select **Launch through EC2** and then select **Launch**.

   Use the EC2 Console to launch the instance and attach an IAM role. This is not possible with the **Launch from Website** action.

7. Follow the prompts to configure and deploy the instance:

   - **Name and tags**: Enter a name and tags for the instance.
   - **Application and OS Images**: Skip this section. The Console agent AMI is already selected.
   - **Instance type**: Depending on region availability, choose an instance type that meets RAM and CPU requirements (t3.2xlarge is preselected and recommended).
   - **Key pair (login)**: Select the key pair that you want to use to securely connect to the instance.
   - **Network settings**: Edit the network settings as needed:
     - Choose the desired VPC and subnet.
     - Specify whether the instance should have a public IP address.
     - Specify security group settings that enable the required connection methods for the Console agent instance: SSH, HTTP, and HTTPS.

       View security group rules for AWS.

   - **Configure storage**: Keep the default size and disk type for the root volume.

     If you want to enable Amazon EBS encryption on the root volume, select **Advanced**, expand **Volume 1**, select **Encrypted**, and then choose a KMS key.

   - **Advanced details**: Under **IAM instance profile**, choose the IAM role that includes the required permissions for the Console agent.
   - **Summary**: Review the summary and select **Launch instance**.

AWS launches the Console agent with the specified settings, and the Console agent runs in about ten minutes.

If the installation fails, you can view logs and a report to help you troubleshoot. Learn how to troubleshoot installation issues.

8. Open a web browser from a host that has a connection to the Console agent virtual machine and URL of the Console agent.
9. After you log in, set up the Console agent:
   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.

      Keep restricted mode disabled to use the Console in standard mode. You should enable restricted mode only if you have a secure environment and want to disconnect this account from the Console backend services. If that's the case, follow steps to get started with NetApp Console in restricted mode.

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

Result

The Console agent is now installed and set up with your Console organization.

Open a web browser and go to the NetApp Console to start using the Console agent with the Console.

If you have Amazon S3 buckets in the same AWS account where you created the Console agent, you'll see an Amazon S3 working environment appear on the **Systems** page automatically. Learn how to manage S3 buckets from NetApp Console

Manually install the Console agent in AWS

You can manually install a Console agent on a Linux host running in AWS. To manually install the Console agent on your own Linux host, you need to review host requirements, set up your networking, prepare AWS permissions, install the Console agent, and then provide the permissions that you prepared.

Before you begin

- You should have an understanding of Console agents.
- You should review Console agent limitations.

Step 1: Review host requirements

Ensure the host running the Console agent software meets operating system, RAM, and port requirements.

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.

AWS EC2 instance type

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

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				

Operating system	Supported OS versions	Supported agent versions	Required container tool	SELinux
	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
	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

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

Key pair

When you create the Console agent, you'll need to select an EC2 key pair to use with the instance.

PUT response hop limit when using IMDSv2

If IMDSv2 is enabled (the default for new EC2 instances), set the PUT response hop limit to 3. If you do not, the system displays a UI initialization error during agent setup.

- Require the use of IMDSv2 on Amazon EC2 instances
- AWS documentation: Change the PUT response hop limit

Step 2: 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
```

Step 3: Set up networking

Ensure the network location supports the following requirements so the Console agent can manage resources in your hybrid cloud.

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.

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.

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

Step 4: Set up AWS permissions for the Console

Provide AWS permissions to the NetApp Console using one of these options:

- Option 1: Create IAM policies and attach the policies to an IAM role that you can associate with the EC2 instance.
- Option 2: Provide the Console with the AWS access key for an IAM user who has the required permissions.

Follow the steps to prepare permissions for the Console.

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 you plann 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. 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 that you can associate with the EC2 instance after you install the Console agent.

AWS access key

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 plan 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 now have an IAM user that has the required permissions and an access key that you can provide to the Console.

Step 5: Install the Console agent

After you complete the prerequisites, manually install the software on your Linux host.

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.
2. Wait for the installation to complete.

   At the end of the installation, the Console agent service (occm) restarts twice if you specified a proxy server.

   If the installation fails, you can view the installation report and logs to help you fix the issues. Learn how to troubleshoot installation issues.

1. Open a web browser from a host that has a connection to the Console agent virtual machine and enter the following URL:

   https://*ipaddress*

2. After you log in, set up the Console agent:
   a. Specify the 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.

      You should keep restricted mode disabled because these steps describe how to use the Console in standard mode. You should enable restricted mode only if you have a secure environment and want to disconnect this account from backend services. If that's the case, follow steps to get started with the NetApp Console in restricted mode.

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

If you have Amazon S3 buckets in the same AWS account where you created the Console agent, you'll see an Amazon S3 storage system appear on the **Systems** page automatically. Learn how to manage S3 buckets

from NetApp ConsoleP

Step 6: Provide permissions to NetApp Console

After you install the Console agent, provide the AWS permissions you set up so the Console agent can manage your data and storage infrastructure in AWS.

IAM role

Attach the IAM role you create 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**.

Go to the NetApp Console to start using the Console agent.

AWS access key

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

Steps

1. Ensure that the correct Console agent is currently selected in the Console.
2. Select **Administration > Credentials**.
3. Select **Organization credentials**.
4. 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**.

Go to the NetApp Console to start using the Console agent.

Azure

Console agent installation options in Azure

There are a few different ways to create a Console agent in Azure. Directly from the NetApp Console is the most common way.

The following installation options are available:

- Create a Console agent directly from the NetApp Console (this is the standard option)

This action launches a VM running Linux and the Console agent software in a VNet of your choice.

- Create a Console agent from the Azure Marketplace

  This action also launches a VM running Linux and the Console agent software, but the deployment is initiated directly from the Azure Marketplace, rather than from the Console.

- Download and manually install the software on your own Linux host

The installation option that you choose impacts how you prepare for installation. This includes how you provide the Console agent with the required permissions that it needs to authenticate and manage resources in Azure.

Create a Console agent in Azure from NetApp Console

To create a Console agent in Azure from the NetApp Console, you need to set up your networking, prepare Azure permissions, and then create the Console agent.

Before you begin

- You should have an understanding of Console agents.
- You should review Console agent limitations.

Step 1: Set up networking

Ensure that the network location where you plan to install the Console agent supports the following requirements. These requirements allow the Console agent to manage hybrid cloud resources.

Azure region

If you use Cloud Volumes ONTAP, the Console agent should be deployed in the same Azure region as the Cloud Volumes ONTAP systems that it manages, or in the Azure region pair for the Cloud Volumes ONTAP systems. This requirement ensures that an Azure Private Link connection is used between Cloud Volumes ONTAP and its associated storage accounts.

Learn how Cloud Volumes ONTAP uses an Azure Private Link

VNet and subnet

When you create the Console agent, you need to specify the VNet and subnet where it should reside.

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 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
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.
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.

Endpoints contacted from the NetApp console

As you use the web-based NetApp Console that's provided through the SaaS layer, it contacts several endpoints to complete data management tasks. This includes endpoints that are contacted to deploy the Console agent from the the Console.

View the list of endpoints contacted from the NetApp console.

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

You need to implement this networking requirement after you create the Console agent.

Step 2: Create a Console agent deployment policy (custom role)

You need to create a custom role that has permissions to deploy the Console agent in Azure.

Create an Azure custom role that you can assign to your Azure account or to a Microsoft Entra service principal. The Console authenticates with Azure and uses these permissions to create the Console agent on your behalf.

The Console deploys the Console agent VM in Azure, enables a system-assigned managed identity, creates the required role, and assigns it to the VM. Review how the Console uses the permissions.

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. Copy the required permissions for a new custom role in Azure and save them in a JSON file.

This custom role contains only the permissions needed to launch the Console agent VM in Azure from the Console. Don't use this policy for other situations. When the Console creates the Console agent, it applies a new set of permissions to the Console agent VM that enables the Console agent to manage Azure resources.

```
{
  "Name": "Azure SetupAsService",
  "Actions": [
    "Microsoft.Compute/disks/delete",
    "Microsoft.Compute/disks/read",
    "Microsoft.Compute/disks/write",
    "Microsoft.Compute/locations/operations/read",
    "Microsoft.Compute/operations/read",
    "Microsoft.Compute/virtualMachines/instanceView/read",
    "Microsoft.Compute/virtualMachines/read",
    "Microsoft.Compute/virtualMachines/write",
    "Microsoft.Compute/virtualMachines/delete",
    "Microsoft.Compute/virtualMachines/extensions/write",
    "Microsoft.Compute/virtualMachines/extensions/read",
    "Microsoft.Compute/availabilitySets/read",
    "Microsoft.Network/locations/operationResults/read",
    "Microsoft.Network/locations/operations/read",
    "Microsoft.Network/networkInterfaces/join/action",
    "Microsoft.Network/networkInterfaces/read",
    "Microsoft.Network/networkInterfaces/write",
    "Microsoft.Network/networkInterfaces/delete",
    "Microsoft.Network/networkSecurityGroups/join/action",
    "Microsoft.Network/networkSecurityGroups/read",
    "Microsoft.Network/networkSecurityGroups/write",
    "Microsoft.Network/virtualNetworks/checkIpAddressAvailability/read",
    "Microsoft.Network/virtualNetworks/read",
    "Microsoft.Network/virtualNetworks/subnets/join/action",
    "Microsoft.Network/virtualNetworks/subnets/read",
    "Microsoft.Network/virtualNetworks/subnets/virtualMachines/read",
    "Microsoft.Network/virtualNetworks/virtualMachines/read",
    "Microsoft.Network/publicIPAddresses/write",
    "Microsoft.Network/publicIPAddresses/read",
    "Microsoft.Network/publicIPAddresses/delete",
    "Microsoft.Network/networkSecurityGroups/securityRules/read",
    "Microsoft.Network/networkSecurityGroups/securityRules/write",
    "Microsoft.Network/networkSecurityGroups/securityRules/delete",
    "Microsoft.Network/publicIPAddresses/join/action",

"Microsoft.Network/locations/virtualNetworkAvailableEndpointServices/rea
d",
    "Microsoft.Network/networkInterfaces/ipConfigurations/read",
```

```
    "Microsoft.Resources/deployments/operations/read",
    "Microsoft.Resources/deployments/read",
    "Microsoft.Resources/deployments/delete",
    "Microsoft.Resources/deployments/cancel/action",
    "Microsoft.Resources/deployments/validate/action",
    "Microsoft.Resources/resources/read",
    "Microsoft.Resources/subscriptions/operationresults/read",
    "Microsoft.Resources/subscriptions/resourceGroups/delete",
    "Microsoft.Resources/subscriptions/resourceGroups/read",
    "Microsoft.Resources/subscriptions/resourcegroups/resources/read",
    "Microsoft.Resources/subscriptions/resourceGroups/write",
    "Microsoft.Authorization/roleDefinitions/write",
    "Microsoft.Authorization/roleAssignments/write",

"Microsoft.MarketplaceOrdering/offertypes/publishers/offers/plans/agreements/read",

"Microsoft.MarketplaceOrdering/offertypes/publishers/offers/plans/agreements/write",
    "Microsoft.Network/networkSecurityGroups/delete",
    "Microsoft.Storage/storageAccounts/delete",
    "Microsoft.Storage/storageAccounts/write",
    "Microsoft.Resources/deployments/write",
    "Microsoft.Resources/deployments/operationStatuses/read",
    "Microsoft.Authorization/roleAssignments/read"
  ],
  "NotActions": [],
  "AssignableScopes": [],
  "Description": "Azure SetupAsService",
  "IsCustom": "true"
}
```

2. Modify the JSON by adding your Azure subscription ID to the assignable scope.

   Example

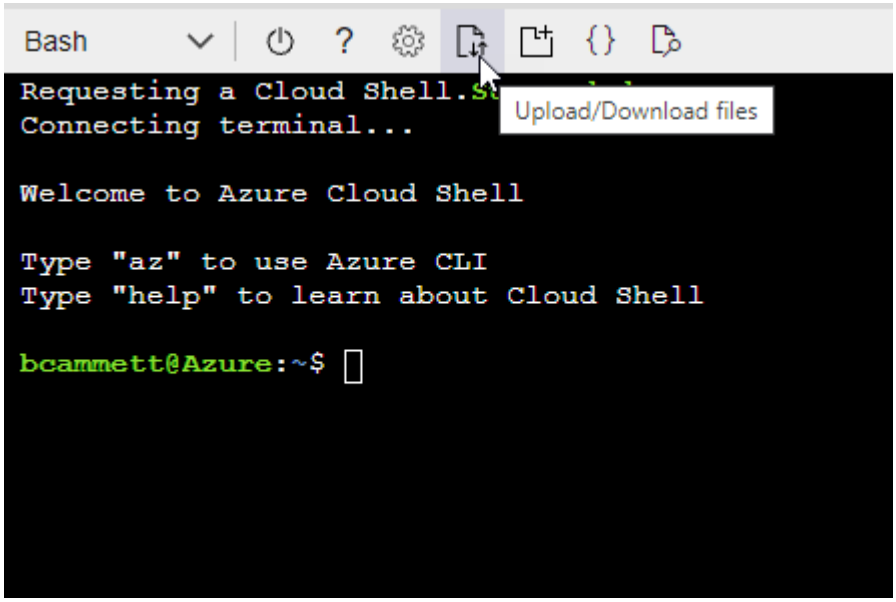
```
"AssignableScopes": [
"/subscriptions/d333af45-0d07-4154-943d-c25fbzzzzzzz"
]
```

3. 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. Enter the following Azure CLI command:

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

You now have a custom role called *Azure SetupAsService*. You can apply this custom role to your user account or to a service principal.

Step 3: Set up authentication

When creating the Console agent from the Console, you need to provide a login that enables the Console to authenticate with Azure and deploy the VM. You have two options:

1. Sign in with your Azure account when prompted. This account must have specific Azure permissions. This is the default option.
2. Provide details about a Microsoft Entra service principal. This service principal also requires specific permissions.

Follow the steps to prepare one of these authentication methods for use with the Console.

Azure account

Assign the custom role to the user who will deploy the Console agent from the Console.

Steps

1. In the Azure portal, open the **Subscriptions** service and select the user's subscription.
2. Click **Access control (IAM)**.
3. Click **Add** > **Add role assignment** and then add the permissions:
   a. Select the **Azure SetupAsService** role and click **Next**.

   > Azure SetupAsService is the default name provided in the Console agent deployment policy for Azure. If you chose a different name for the role, then select that name instead.

   b. Keep **User, group, or service principal** selected.
   c. Click **Select members**, choose your user account, and click **Select**.
   d. Click **Next**.
   e. Click **Review + assign**.

Service principal

Rather than logging in with your Azure account, you can provide the Console with the credentials for an Azure service principal that has the required permissions.

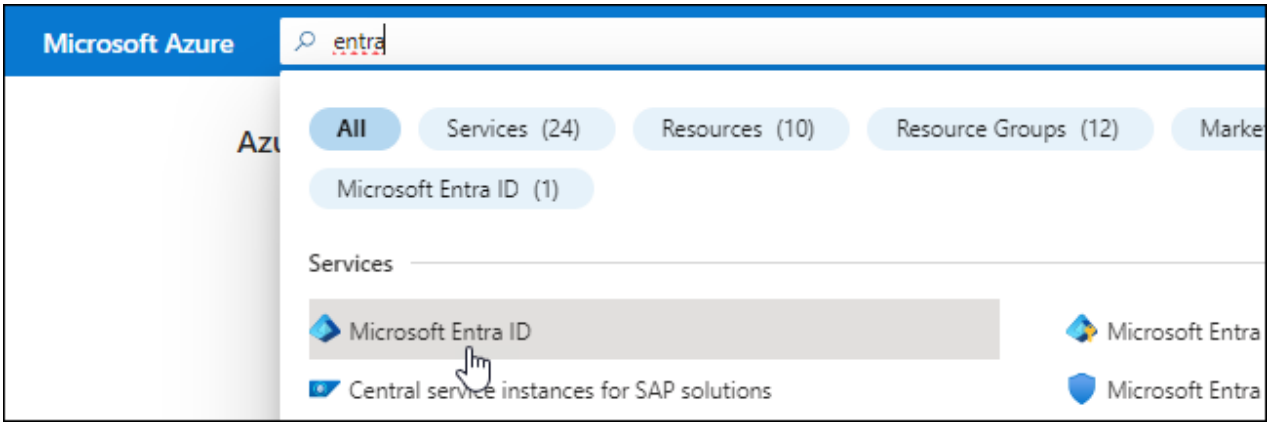
Create and set up a service principal in Microsoft Entra ID and obtain the Azure credentials that the Console needs.

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.



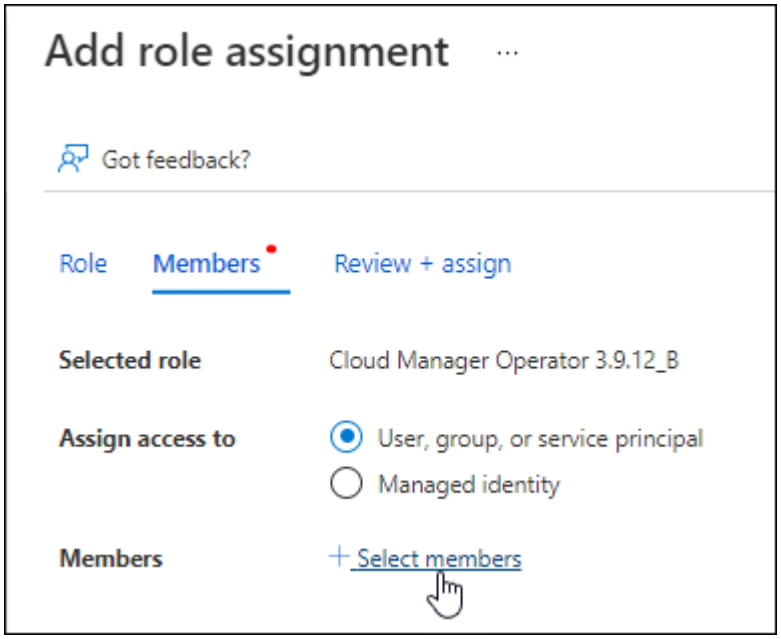
3. In the menu, select **App registrations**.
4. Select **New registration**.

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.

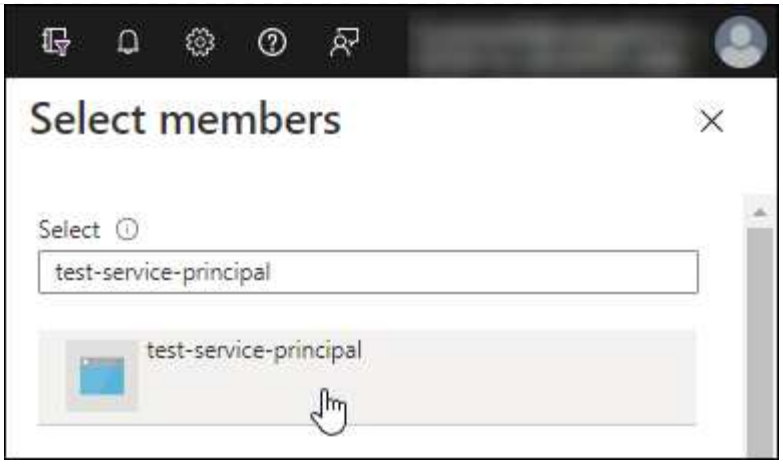
Assign the custom role to the application

1. From the Azure portal, open the **Subscriptions** service.
2. Select the subscription.
3. Click **Access control (IAM) > Add > Add role assignment**.
4. In the **Role** tab, select the **Console Operator** role and click **Next**.
5. In the **Members** tab, complete the following steps:
   a. Keep **User, group, or service principal** selected.
   b. Click **Select members**.



   c. Search for the name of the application.

      Here's an example:



d. Select the application and click **Select**.

e. Click **Next**.

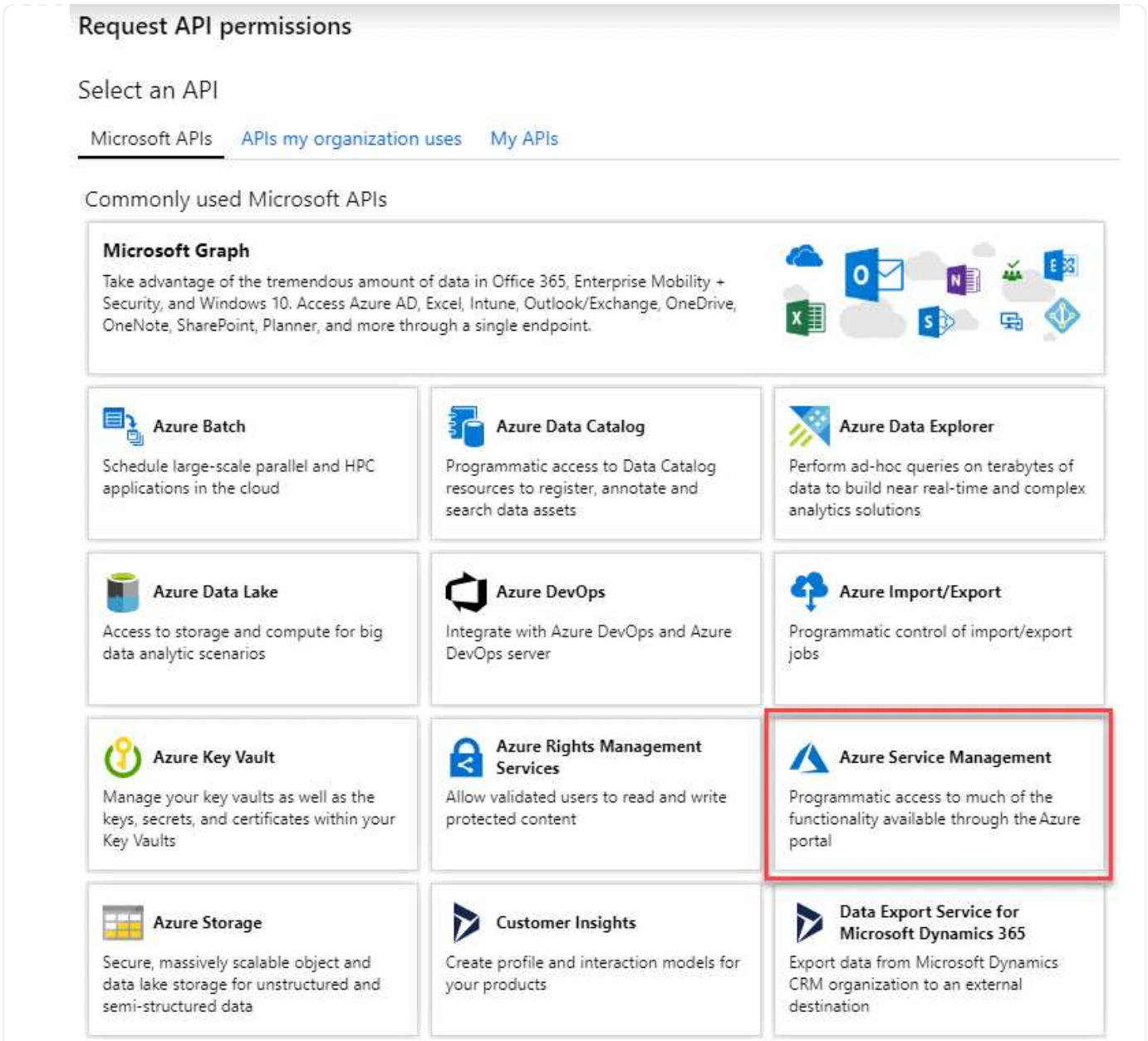
6. Click **Review + assign**.

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

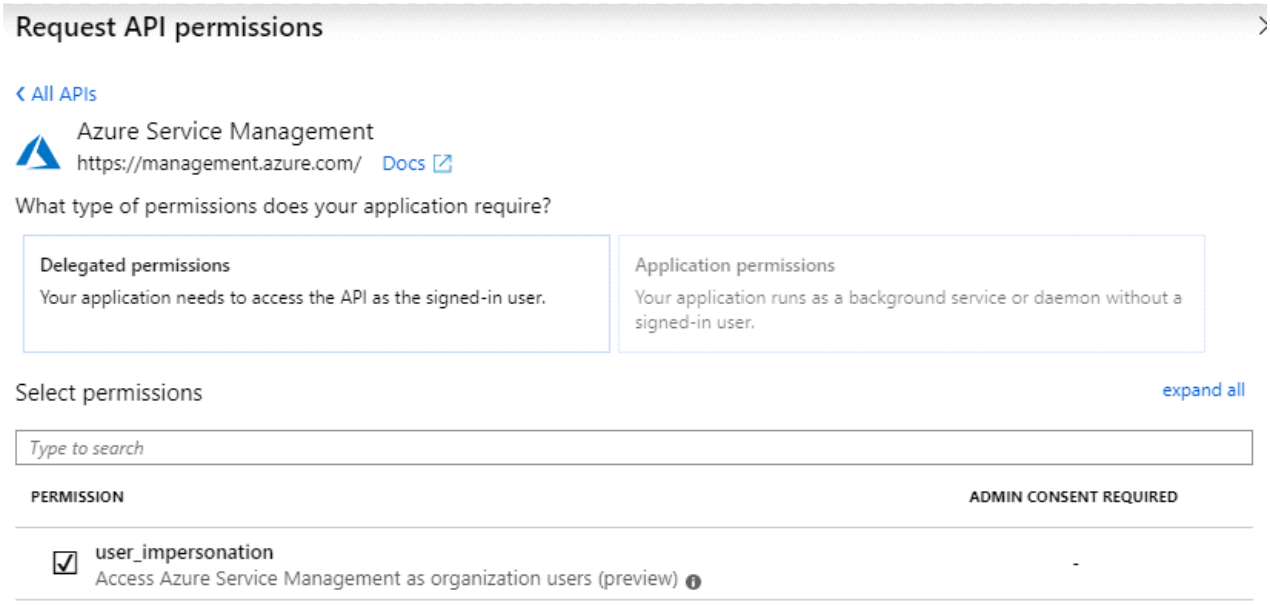
   If you want to manage resources in multiple Azure subscriptions, then you must bind the service principal to each of those subscriptions. For example, the Console enables you to select the subscription that you want to use when deploying Cloud Volumes ONTAP.

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**.

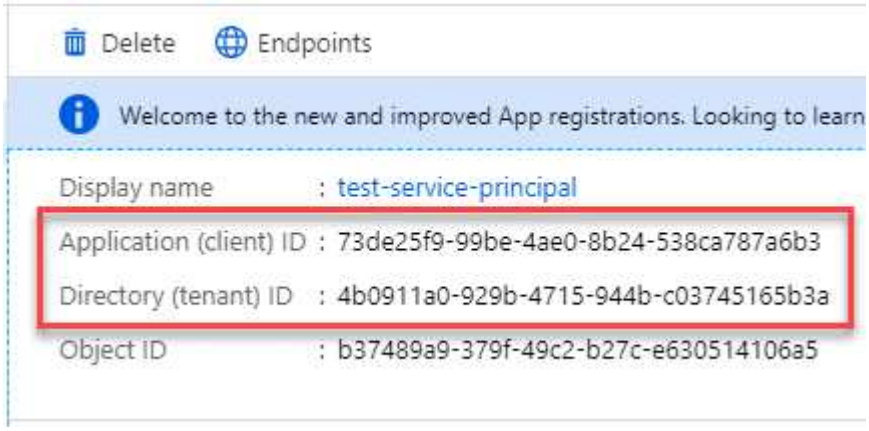


4. Select **Access Azure Service Management as organization users** and then select **Add permissions**.



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. You need to enter this information in the Console when you create the Console agent.

Step 4: Create the Console agent

Create the Console agent directly from the NetApp Console.

About this task

- Creating the Console agent from the Console deploys a virtual machine in Azure using a default configuration. Do not switch to a smaller VM instance with fewer CPUs or less RAM after creating the Console agent. Learn about the default configuration for the Console agent.
- When the Console deploys the Console agent, it creates a custom role and assigns it to the Console agent VM. This role includes permissions that enables the Console agent to manage Azure resources. You need to ensure that the role is kept up to date as new permissions are added in subsequent releases. Learn more about the custom role for the Console agent.

Before you begin

You should have the following:

- An Azure subscription.
- A VNet and subnet in your Azure region of choice.
- Details about a proxy server, if your organization requires a proxy for all outgoing internet traffic:
  - IP address
  - Credentials
  - HTTPS certificate
- An SSH public key, if you want to use that authentication method for the Console agent virtual machine. The other option for the authentication method is to use a password.

  Learn about connecting to a Linux VM in Azure

- If you don’t want the Console to automatically create an Azure role for the Console agent, then you’ll need to create your own using the policy on this page.

  These permissions are for the Console agent itself. It’s a different set of permissions than what you previously set up to deploy the Console agent VM.

Steps

1. Select **Administration > Agents**.
2. On the **Overview** page, select **Deploy agent > Azure**
3. On the **Review** page, review the requirements for deploying an agent. Those requirements are also detailed above on this page.
4. On the **Virtual Machine Authentication** page, select the authentication option that matches how you set up Azure permissions:
   - Select **Log in** to log in to your Microsoft account, which should have the required permissions.

     The form is owned and hosted by Microsoft. Your credentials are not provided to NetApp.

     If you're already logged in to an Azure account, then the Console automatically uses that account. If you have multiple accounts, then you might need to log out first to ensure that you're using the right account.

   - Select **Active Directory service principal** to enter information about the Microsoft Entra service principal that grants the required permissions:
     - Application (client) ID
     - Directory (tenant) ID
     - Client Secret

   Learn how to obtain these values for a service principal.
5. On the **Virtual Machine Authentication** page, choose an Azure subscription, a location, a new resource group or an existing resource group, and then choose an authentication method for the Console agent virtual machine that you're creating.

   The authentication method for the virtual machine can be a password or an SSH public key.

   Learn about connecting to a Linux VM in Azure
6. On the **Details** page, enter a name for the agent, specify tags, and choose whether you want the Console to create a new role that has the required permissions, or if you want to select an existing role that you set up with the required permissions.

   Note that you can choose the Azure subscriptions associated with this role. Each subscription that you choose provides the Console agent permissions to manage resources in that subscription (for example, Cloud Volumes ONTAP).
7. On the **Network** page, choose a VNet and subnet, whether to enable a public IP address, and optionally specify a proxy configuration.
   - On the **Security Group** page, choose whether to create a new security group or whether to select an existing security group that allows the required inbound and outbound rules.

     View security group rules for Azure.
8. Review your selections to verify that your set up is correct.
   a. The **Validate agent configuration** check box is marked by default to have the Console validate the network connectivity requirements when you deploy. If the Console fails to deploy the agent, it provides a report to help you troubleshoot. If the deployment succeeds, no report is provided.

If you are still using the previous endpoints used for agent upgrades, the validation fails with an error. To avoid this, unmark the check box to skip the validation check.

9. Select **Add**.

   The Console prepares the agent in about 10 minutes. Stay on the page until the process completes.

Result

After the process is complete, the Console agent is available for use from the Console.

If the deployment fails, you can download a report and logs from the Console to help you fix the issues. Learn how to troubleshoot installation issues.

If you have Azure Blob storage in the same Azure account where you created the Console agent, you'll see Azure Blob storage appear on the **Systems** page automatically. Learn how to manage Azure Blob storage from NetApp Console

Create a Console agent from the Azure Marketplace

You can create a Console agent in Azure directly from the Azure Marketplace. To create a Console agent from the Azure Marketplace, you need to set up your networking, prepare Azure permissions, review instance requirements, and then create the Console agent.

Before you begin

- You should have an understanding of Console agents.
- Review Console agent limitations.

Step 1: Set up networking

Ensure that the network location where you plan to install the Console agent supports the following requirements.These requirements enable the Console agent to manage resources in your hybrid cloud.

Azure region

If you use Cloud Volumes ONTAP, the Console agent should be deployed in the same Azure region as the Cloud Volumes ONTAP systems that it manages, or in the Azure region pair for the Cloud Volumes ONTAP systems. This requirement ensures that an Azure Private Link connection is used between Cloud Volumes ONTAP and its associated storage accounts.

Learn how Cloud Volumes ONTAP uses an Azure Private Link

VNet and subnet

When you create the Console agent, you need to specify the VNet and subnet where it should reside.

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 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
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.
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.

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

Implement the networking requirements after creating the Console agent.

Step 2: Review VM requirements

When you create the Console agent, choose a virtual machine type that meets the following requirements.

CPU

8 cores or 8 vCPUs

RAM

32 GB

Azure VM size

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

Step 3: Set up permissions

You can provide permissions in the following ways:

- Option 1: Assign a custom role to the Azure VM using a system-assigned managed identity.
- Option 2: Provide the Console with the credentials for an Azure service principal that has the required permissions.

Follow these steps to set up permissions for the Console.

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

Steps

1. If you're planning to manually install the software on your own host, enable a system-assigned managed identity on the VM 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 Connector 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

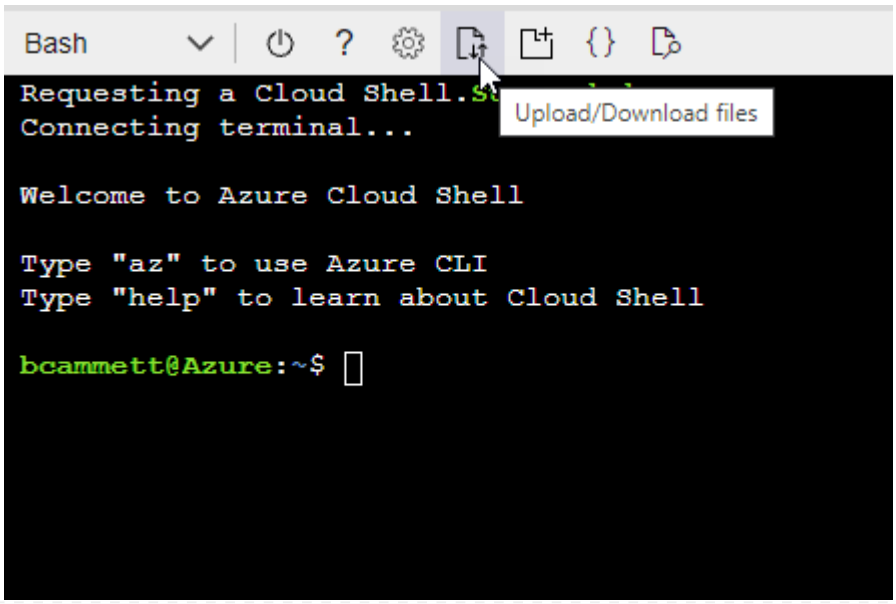
   ```
   "AssignableScopes": [
   "/subscriptions/d333af45-0d07-4154-943d-c25fbzzzzzzz",
   "/subscriptions/54b91999-b3e6-4599-908e-416e0zzzzzzz",
   "/subscriptions/398e471c-3b42-4ae7-9b59-ce5bbzzzzzzz"
   ]
   ```

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
```

Service principal

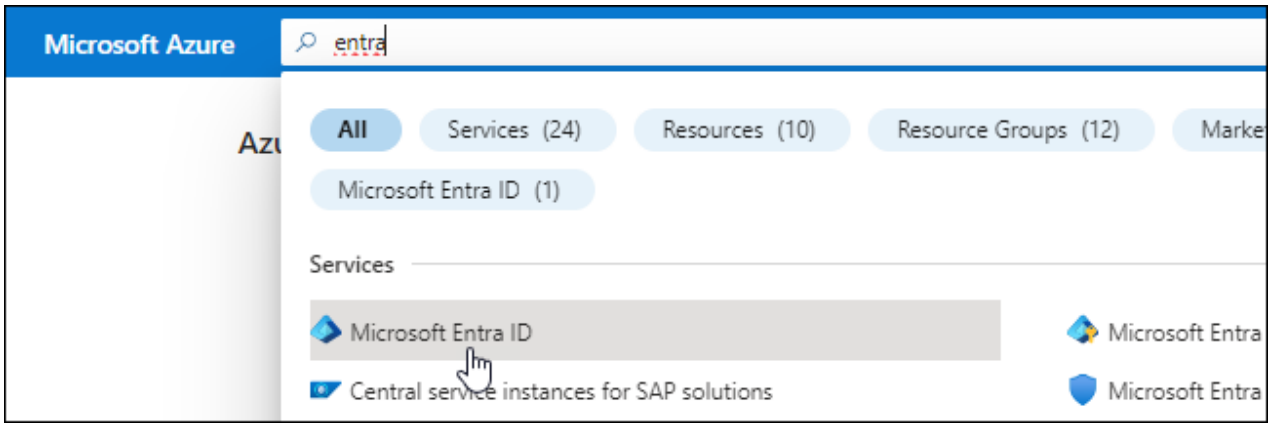
Create and set up a service principal in Microsoft Entra ID and obtain the Azure credentials that the Console needs.

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.

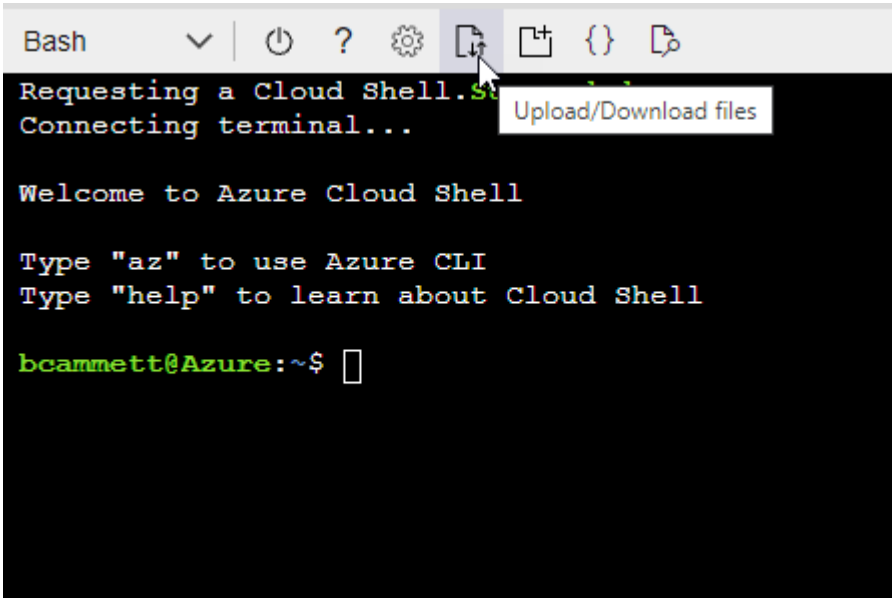
Example

```
"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.

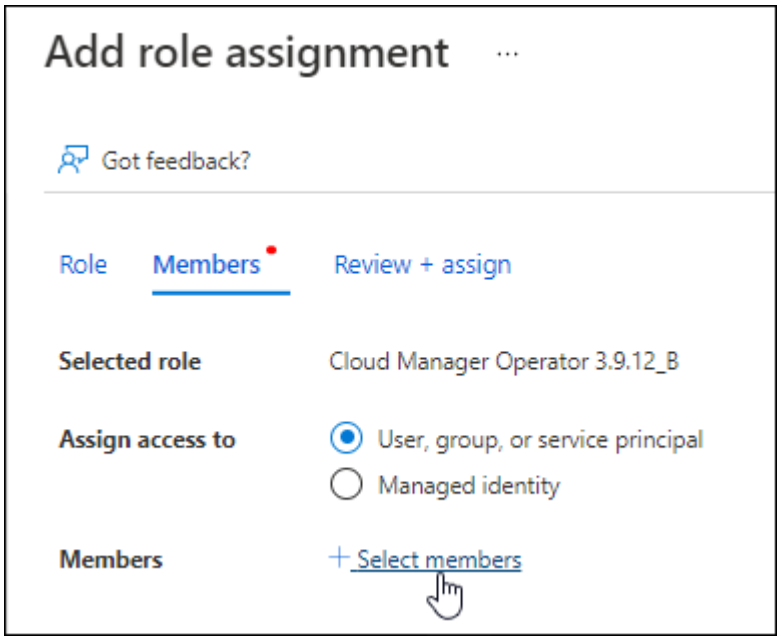
b. Select the subscription.

c. Select **Access control (IAM) > Add > Add role assignment**.

d. In the **Role** tab, select the **Console Operator** role and select **Next**.

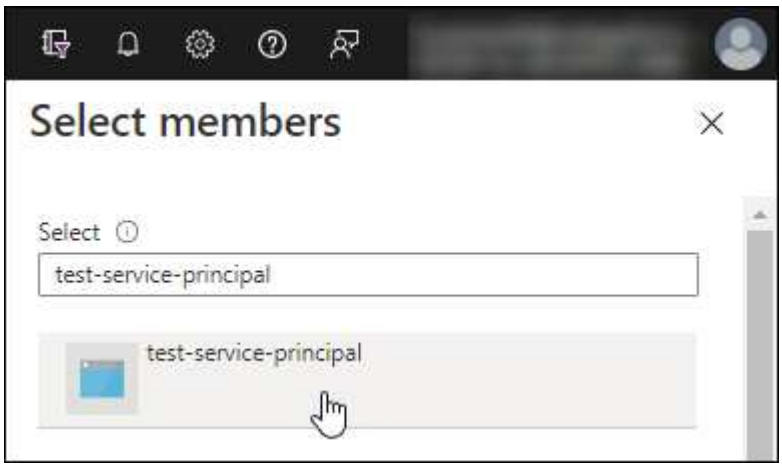
e. In the **Members** tab, complete the following steps:

- 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**.

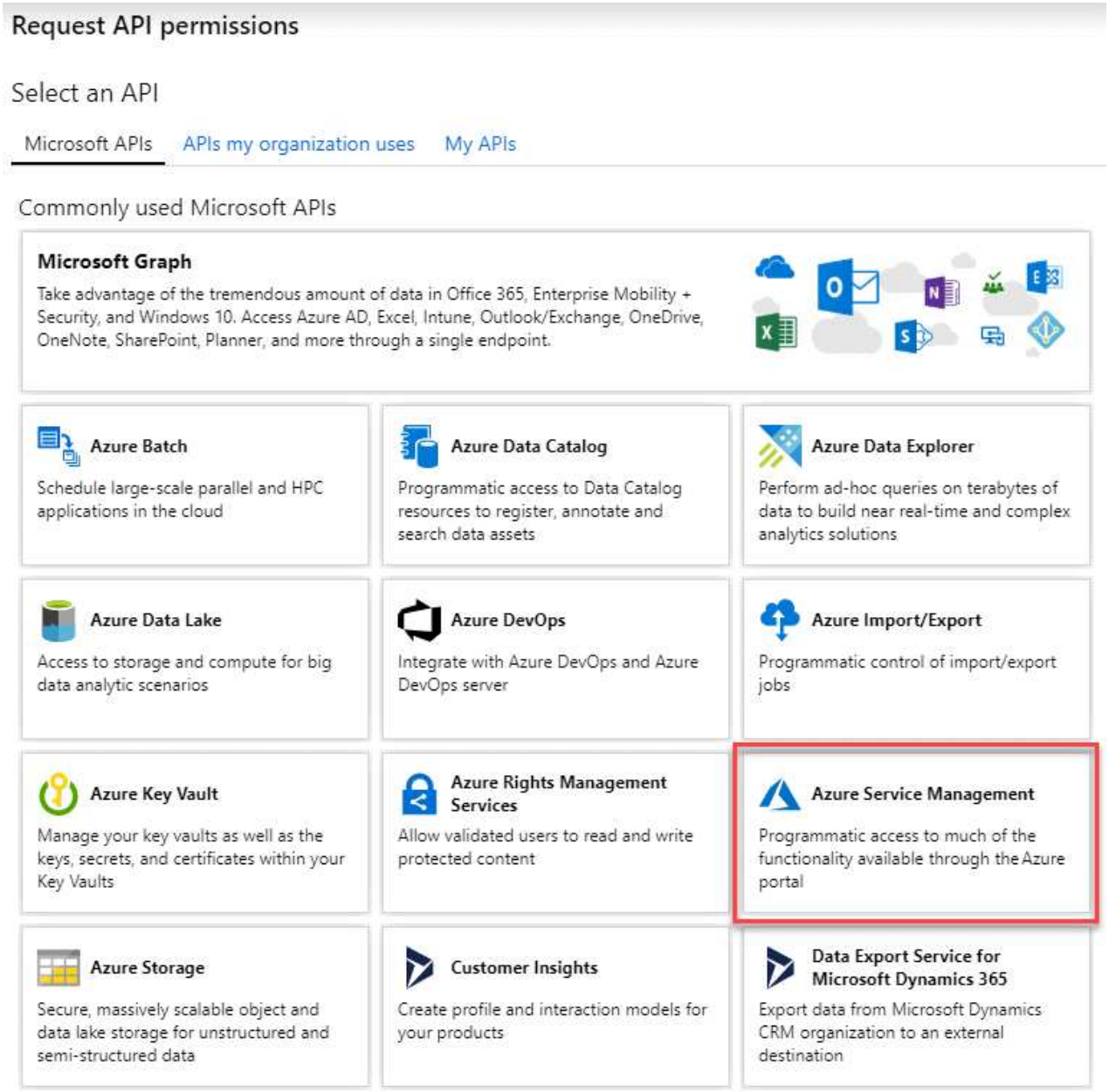
The service principal now has the required Azure permissions to deploy the Console agent.

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

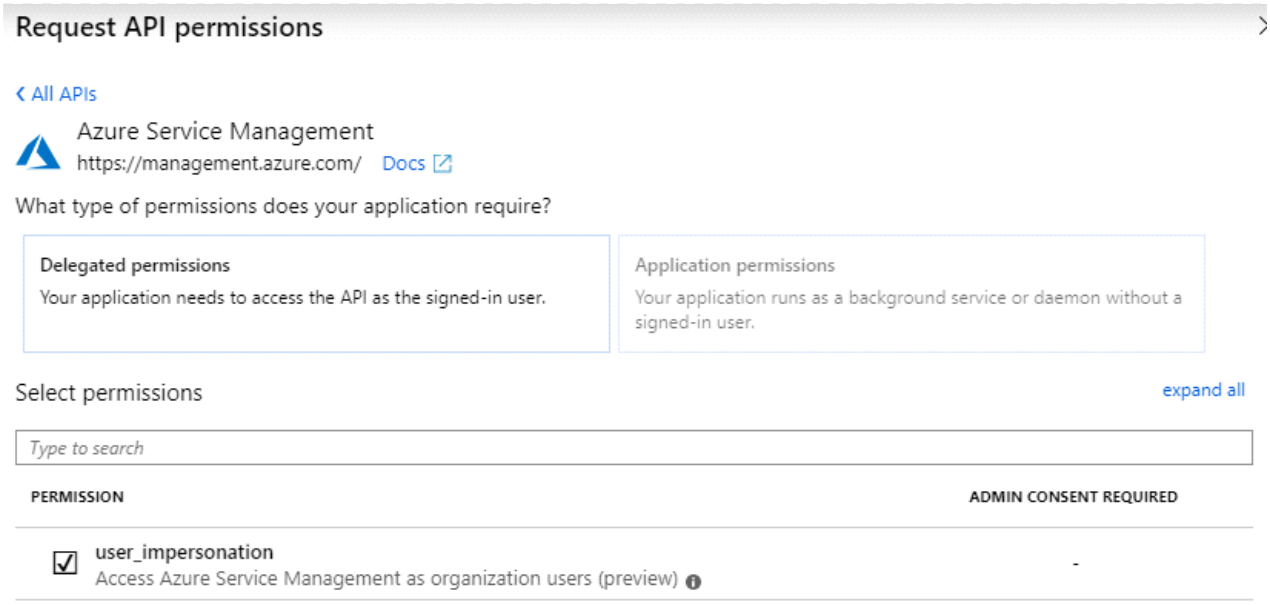
the subscription that you want to use when deploying Cloud Volumes ONTAP.

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**.

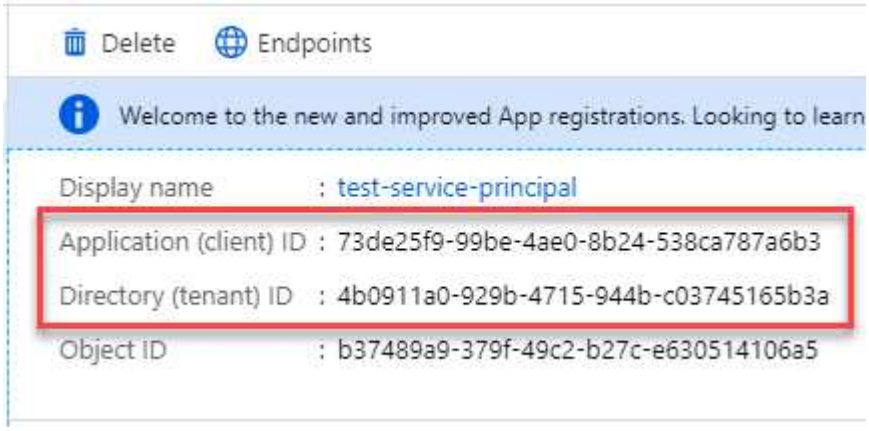


4. Select **Access Azure Service Management as organization users** and then select **Add permissions**.



Get the application ID and directory ID for the application

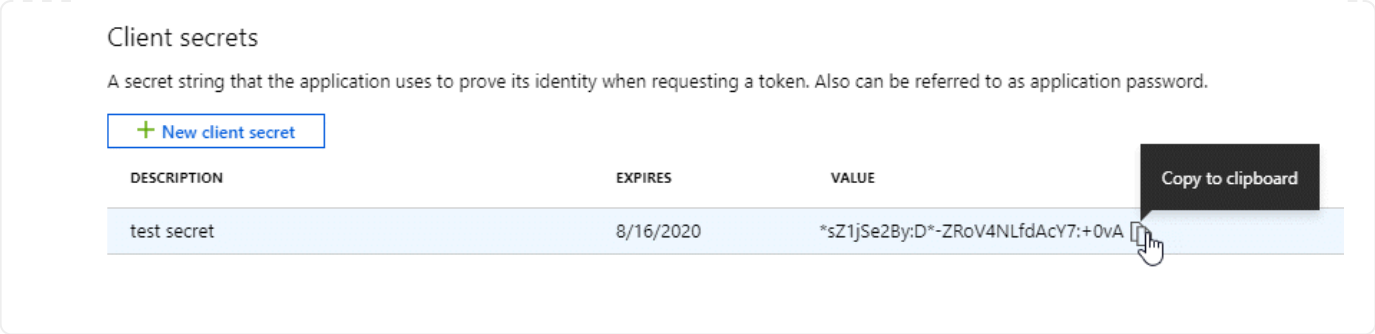
1. In the **Microsoft Entra ID** service, select **App registrations** and select the application.
2. Copy the **Application (client) ID** and the **Directory (tenant) ID**.



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.



Step 4: Create the Console agent

Launch the Console agent directly from the Azure Marketplace.

About this task

Creating the Console agent from the Azure Marketplace sets up a virtual machine with a default configuration. Learn about the default configuration for the Console agent.

Before you begin

You should have the following:

- An Azure subscription.
- A VNet and subnet in your Azure region of choice.
- Details about a proxy server, if your organization requires a proxy for all outgoing internet traffic:
  - IP address
  - Credentials
  - HTTPS certificate
- An SSH public key, if you want to use that authentication method for the Console agent virtual machine. The other option for the authentication method is to use a password.

  Learn about connecting to a Linux VM in Azure

- If you don't want the Console to automatically create an Azure role for the Console agent, then you'll need to create your own using the policy on this page.

  These permissions are for the Console agent instance itself. It's a different set of permissions than what you previously set up to deploy the Console agent VM.

Steps

1. Go to the NetApp Console agent VM page in the Azure Marketplace.

   Azure Marketplace page for commercial regions

2. Select **Get it now** and then select **Continue**.
3. From the Azure portal, select **Create** and follow the steps to configure the virtual machine.

   Note the following as you configure the VM:

   - **VM size**: Choose a VM size that meets CPU and RAM requirements. We recommend Standard_D8s_v3.

- **Disks**: The Console agent can perform optimally with either HDD or SSD disks.
- **Network security group**: The Console agent requires inbound connections using SSH, HTTP, and HTTPS.

  View security group rules for Azure.

- Identity*: Under **Management**, select **Enable system assigned managed identity**.

  This setting is important because a managed identity allows the Console agent virtual machine to identify itself to Microsoft Entra ID without providing any credentials. Learn more about managed identities for Azure resources.

4. On the **Review + create** page, review your selections and select **Create** to start the deployment.

   Azure deploys the virtual machine with the specified settings. You should see the virtual machine and Console agent software running in about ten minutes.

   If the installation fails, you can view logs and a report to help you troubleshoot. Learn how to troubleshoot installation issues.

5. Open a web browser from a host that has a connection to the Console agent virtual machine and enter the following URL:

   https://*ipaddress*

6. After you log in, set up the Console agent:
   a. Specify the 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.

      Keep restricted mode disabled to use the Console in standard mode. You should enable restricted mode only if you have a secure environment and want to disconnect this account from the Console backend services. If that's the case, follow steps to get started with the Console in restricted mode.

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

Result

You have now installed the Console agent and set it up with your the Console organization.

If you have Azure Blob storage in the same Azure subscription where you created the Console agent, you'll see an Azure Blob storage system appear on the **Systems** page automatically. Learn how to manage Azure Blob storage from the Console

Step 5: Provide permissions to the Console agent

Now that you've created the Console agent, you need to provide it with the permissions that you previously set up. Providing the permissions enables the Console agent to manage your data and storage infrastructure in Azure.

Custom 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.

What's next?

Go to the NetApp Console to start using the Console agent.

Service principal

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 now has the permissions that it needs to perform actions in Azure on your behalf.

Manually install the Console agent in Azure

To manually install the Console agent on your own Linux host, you need to review host requirements, set up your networking, prepare Azure permissions, install the Console agent, and then provide the permissions that you prepared.

Before you begin

- You should have an understanding of Console agents.
- You should review Console agent limitations.

Step 1: Review host requirements

The Console agent software must run on a host that meets specific operating system requirements, RAM requirements, port requirements, and so on.

> 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.

Azure VM size

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

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
	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

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

Step 2: 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 2. 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 3: Set up networking

Ensure that the network location where you plan to install the Console agent supports the following requirements. Meeting these requirements enables the Console agent to manage resources and processes within your hybrid cloud environment.

Azure region

If you use Cloud Volumes ONTAP, the Console agent should be deployed in the same Azure region as the

Cloud Volumes ONTAP systems that it manages, or in the Azure region pair for the Cloud Volumes ONTAP systems. This requirement ensures that an Azure Private Link connection is used between Cloud Volumes ONTAP and its associated storage accounts.

Learn how Cloud Volumes ONTAP uses an Azure Private Link

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.

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

Step 4: Set up Console agent deployment permissions

You need to provide Azure permissions to the Console agent by using one of the following options:

- Option 1: Assign a custom role to the Azure VM using a system-assigned managed identity.
- Option 2: Provide the Console agent with the credentials for an Azure service principal that has the required permissions.

Follow the steps to prepare permissions for the Console agent.

Create a custom role for Console agent deployment

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. If you're planning to manually install the software on your own host, enable a system-assigned managed identity on the VM 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 Connector 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

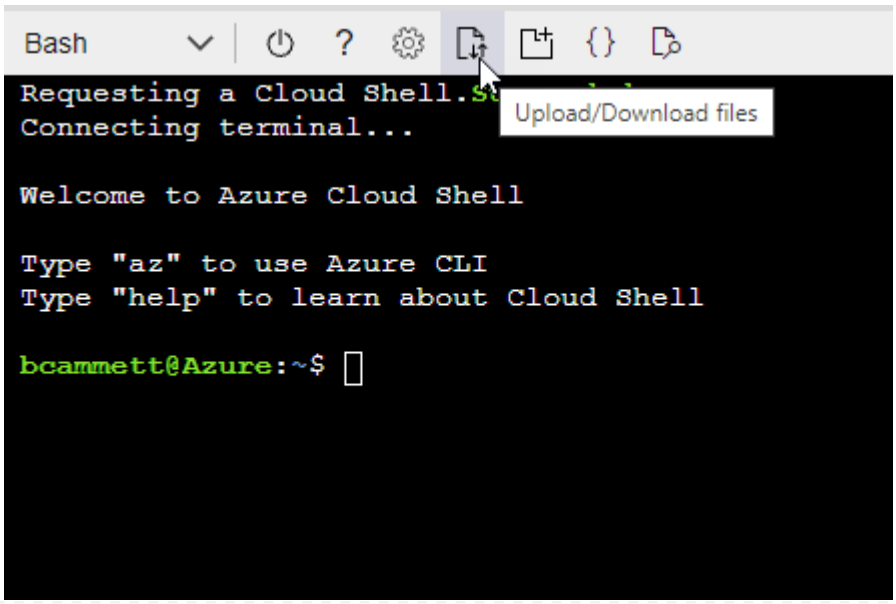
   ```
   "AssignableScopes": [
   "/subscriptions/d333af45-0d07-4154-943d-c25fbzzzzzzz",
   "/subscriptions/54b91999-b3e6-4599-908e-416e0zzzzzzz",
   "/subscriptions/398e471c-3b42-4ae7-9b59-ce5bbzzzzzzz"
   ]
   ```

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
```

Service principal

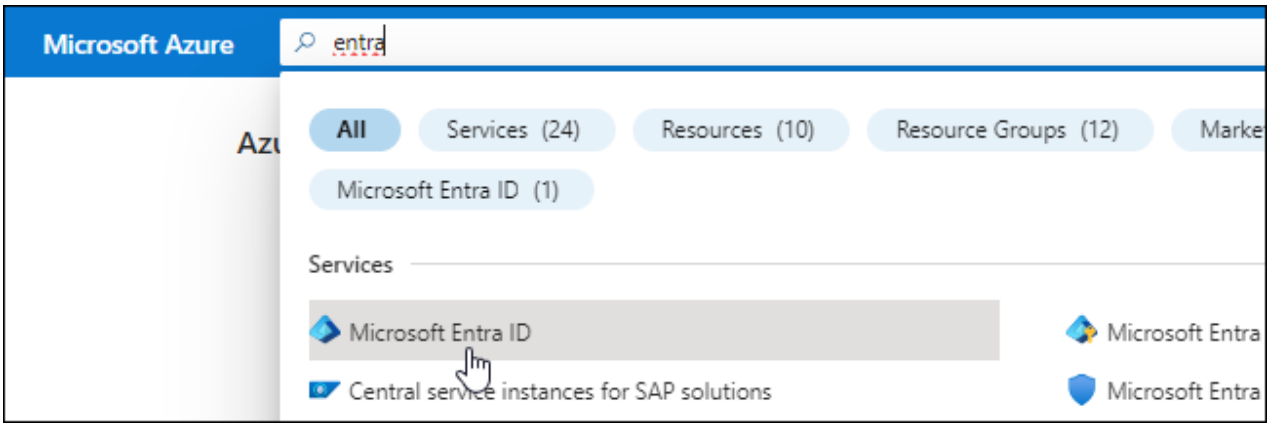
Create and set up a service principal in Microsoft Entra ID and obtain the Azure credentials that the Console agent needs.

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.

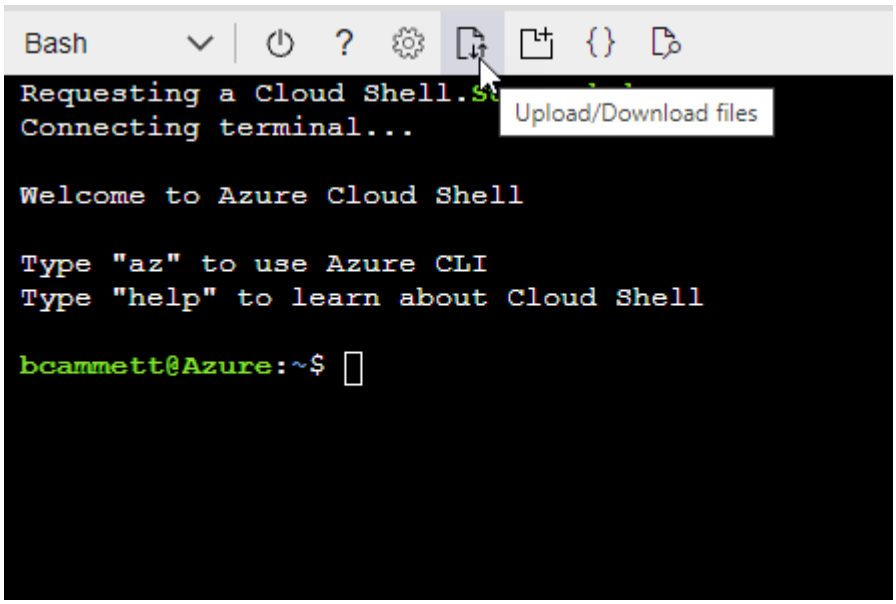
Example

```
"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.

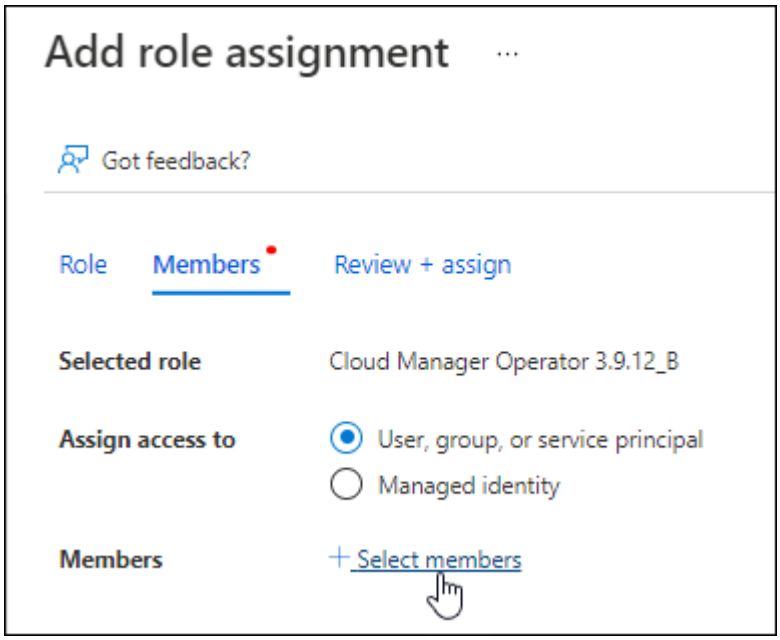
b. Select the subscription.

c. Select **Access control (IAM) > Add > Add role assignment**.

d. In the **Role** tab, select the **Console Operator** role and select **Next**.

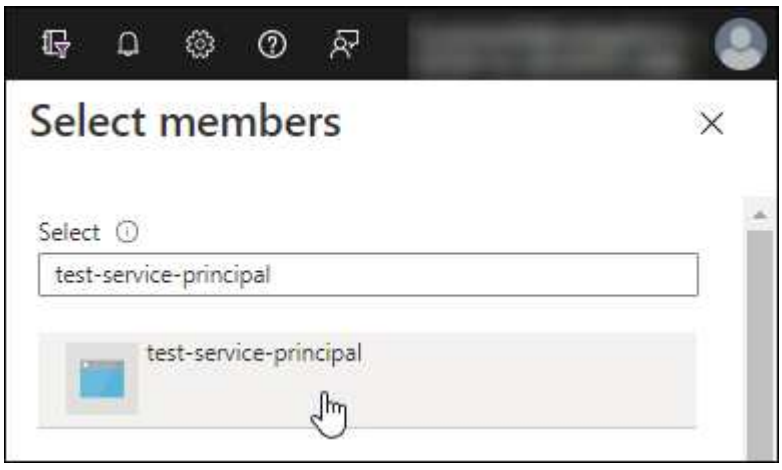
e. In the **Members** tab, complete the following steps:

- 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**.

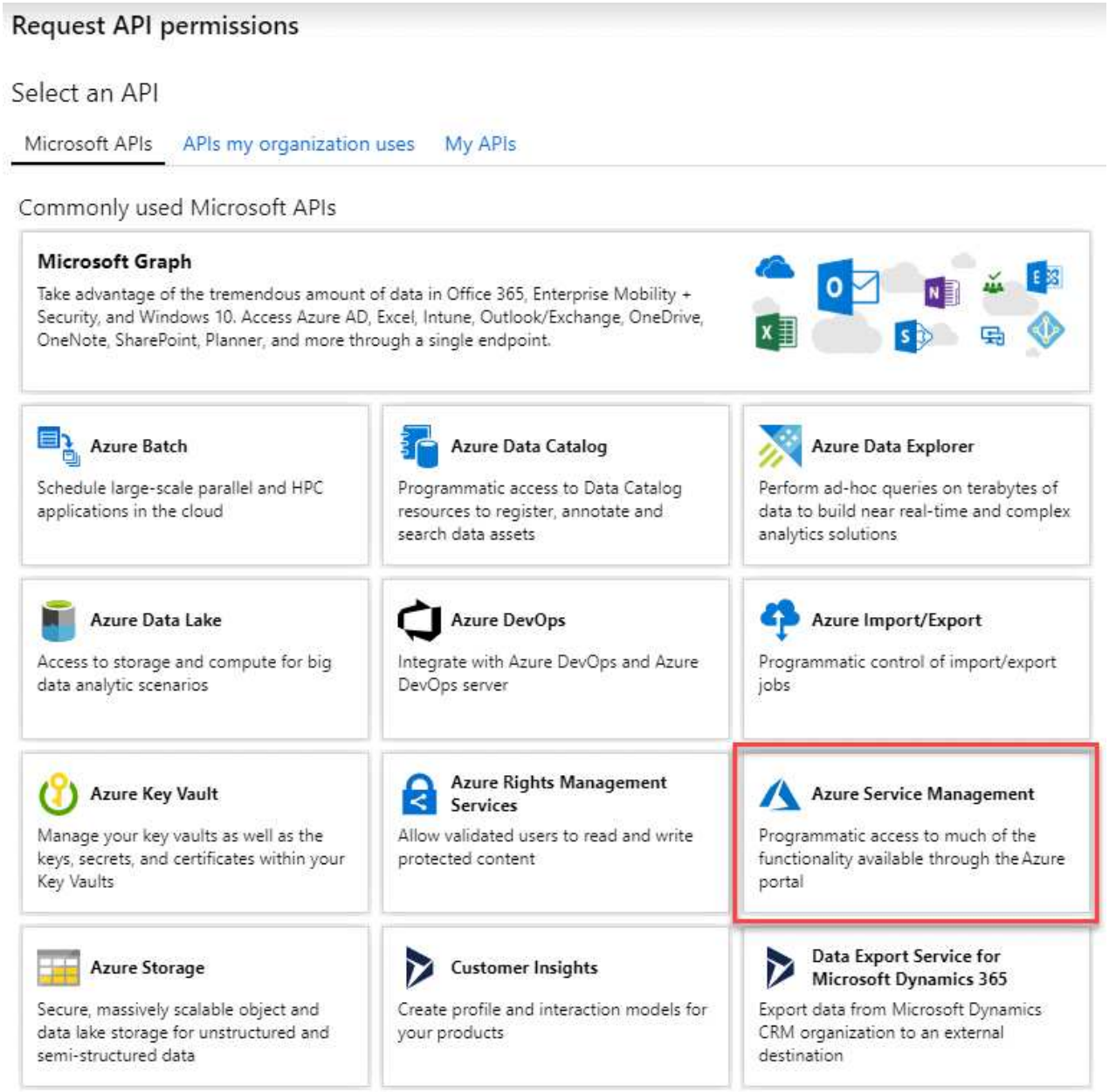
The service principal now has the required Azure permissions to deploy the Console agent.

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

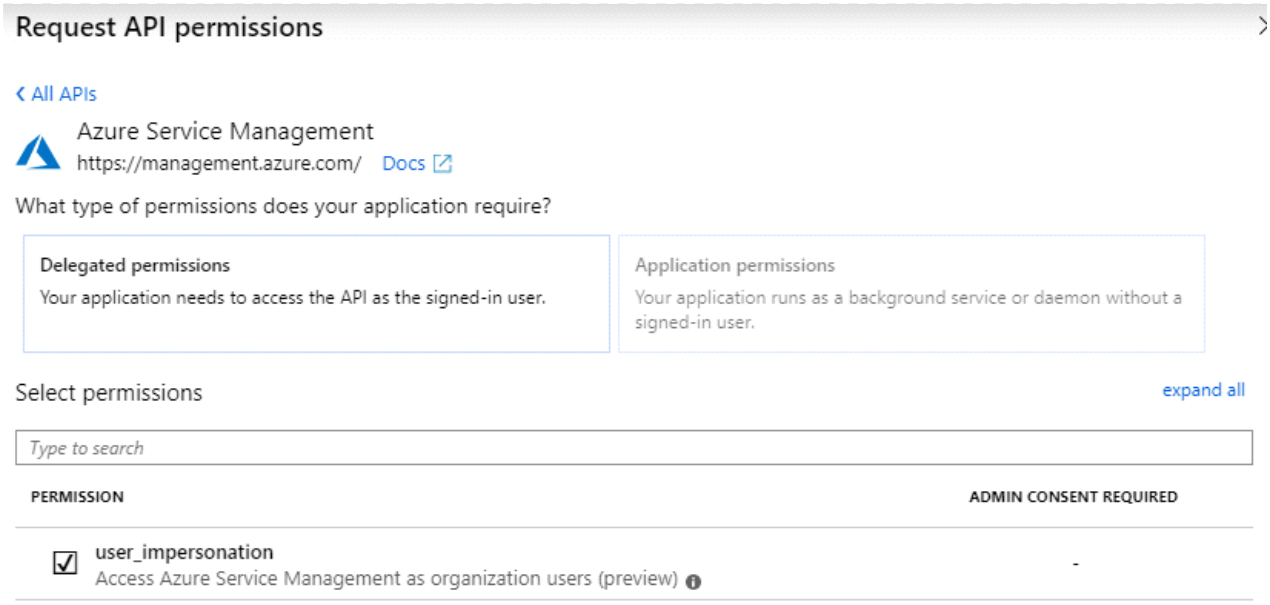
the subscription that you want to use when deploying Cloud Volumes ONTAP.

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**.

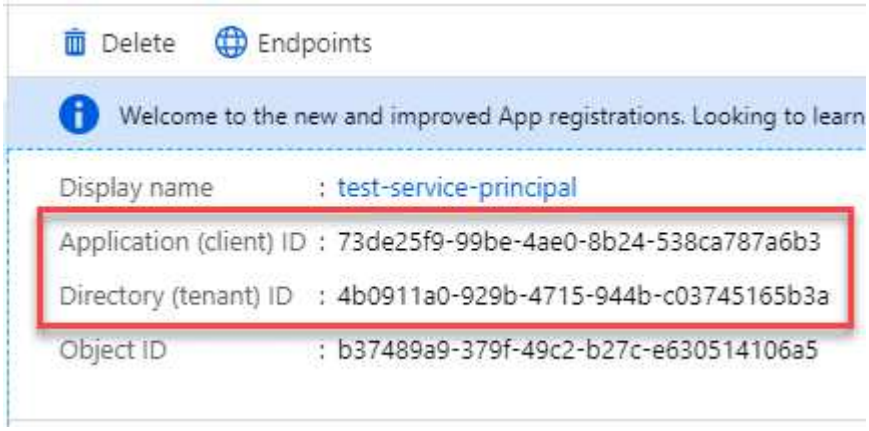


4. Select **Access Azure Service Management as organization users** and then select **Add permissions**.



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.

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. You need to enter this information in the Console when you add an Azure account.

Step 5: Install the Console agent

After the pre-requisites are complete, you can manually install the software on your own Linux host.

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.

- A managed identity enabled on the VM in Azure 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

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.
2. Wait for the installation to complete.

   At the end of the installation, the Console agent service (occm) restarts twice if you specified a proxy server.

   If the installation fails, you can view the installation report and logs to help you fix the issues. Learn how to troubleshoot installation issues.

1. Open a web browser from a host that has a connection to the Console agent virtual machine and enter the following URL:

https://*ipaddress*

2. After you log in, set up the Console agent:
   a. Specify the 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.

      You should keep restricted mode disabled because these steps describe how to use the Console in standard mode. You should enable restricted mode only if you have a secure environment and want to disconnect this account from backend services. If that's the case, follow steps to get started with the NetApp Console in restricted mode.

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

If you have Azure Blob storage in the same Azure subscription where you created the Console agent, you'll see an Azure Blob storage system appear on the **Systems** page automatically. Learn how to manage Azure Blob storage from NetApp Console

Step 6: Provide permissions to NetApp Console

Now that you've installed the Console agent, you need to provide the Console agent with the Azure permissions that you previously set up. Providing the permissions enables the Console to manage your data and storage infrastructure in Azure.

Custom 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.

What's next?

Go to the NetApp Console to start using the Console agent.

Service principal

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.

Google Cloud

Console agent installation options in Google Cloud

There are a few different ways to create a Console agent in Google Cloud. Directly from the NetApp Console is the most common way.

The following installation options are available:

- Create the Console agent directly from the Console (this is the standard option)

  This action launches a VM instance running Linux and the Console agent software in a VPC of your choice.

- Create the Console agent using Google Platform

  This action also launches a VM instance running Linux and the Console agent software, but the deployment is initiated directly from Google Cloud, rather than from the Console.

- Download and manually install the software on your own Linux host

The installation option that you choose impacts how you prepare for installation. This includes how you provide the Console with the required permissions that it needs to authenticate and manage resources in Google Cloud.

Create a Console agent in Google Cloud from NetApp Console

You can create a Console agent in Google Cloud from the Console. You need to set up your networking, prepare Google Cloud permissions, enable Google Cloud APIs, and then create the Console agent.

Before you begin

- You should have an understanding of Console agents.
- You should review Console agent limitations.

Step 1: Set up networking

Set up networking to ensure the Console agent can manage resources, with connections to target networks and outbound internet access.

VPC and subnet

When you create the Console agent, you need to specify the VPC and subnet where it should reside.

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 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
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.
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.

Endpoints contacted from the NetApp console

As you use the web-based NetApp Console that's provided through the SaaS layer, it contacts several endpoints to complete data management tasks. This includes endpoints that are contacted to deploy the Console agent from the the Console.

View the list of endpoints contacted from the NetApp console.

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

Implement this networking requirement after creating the Console agent.

Step 2: Set up permissions to create the Console agent

Before you can deploy a Console agent from the Console, you need to set up permissions for the Google Platform user who deploys the Console agent VM.

Steps

1. Create a custom role in Google Platform:
   a. Create a YAML file that includes the following permissions:

```
title: Console agent deployment policy
description: Permissions for the user who deploys the Console agent
stage: GA
includedPermissions:

- cloudbuild.builds.get
- compute.disks.create
- compute.disks.get
- compute.disks.list
- compute.disks.setLabels
- compute.disks.use
- compute.firewalls.create
- compute.firewalls.delete
- compute.firewalls.get
- compute.firewalls.list
- compute.globalOperations.get
- compute.images.get
- compute.images.getFromFamily
- compute.images.list
- compute.images.useReadOnly
- compute.instances.attachDisk
- compute.instances.create
- compute.instances.get
- compute.instances.list
- compute.instances.setDeletionProtection
- compute.instances.setLabels
- compute.instances.setMachineType
- compute.instances.setMetadata
- compute.instances.setTags
- compute.instances.start
- compute.instances.updateDisplayDevice
```

```
- compute.machineTypes.get
- compute.networks.get
- compute.networks.list
- compute.networks.updatePolicy
- compute.projects.get
- compute.regions.get
- compute.regions.list
- compute.subnetworks.get
- compute.subnetworks.list
- compute.zoneOperations.get
- compute.zones.get
- compute.zones.list
- config.deployments.create
- config.operations.get
- config.deployments.delete
- config.deployments.deleteState
- config.deployments.get
- config.deployments.getState
- config.deployments.list
- config.deployments.update
- config.deployments.updateState
- config.previews.get
- config.previews.list
- config.revisions.get
- config.resources.list
- deploymentmanager.compositeTypes.get
- deploymentmanager.compositeTypes.list
- deploymentmanager.deployments.create
- deploymentmanager.deployments.delete
- deploymentmanager.deployments.get
- deploymentmanager.deployments.list
- deploymentmanager.manifests.get
- deploymentmanager.manifests.list
- deploymentmanager.operations.get
- deploymentmanager.operations.list
- deploymentmanager.resources.get
- deploymentmanager.resources.list
- deploymentmanager.typeProviders.get
- deploymentmanager.typeProviders.list
- deploymentmanager.types.get
- deploymentmanager.types.list
- resourcemanager.projects.get
- compute.instances.setServiceAccount
- iam.serviceAccounts.actAs
- iam.serviceAccounts.create
- iam.serviceAccounts.list
```

```
- iam.serviceAccountKeys.create
- storage.buckets.create
- storage.buckets.get
- storage.objects.create
- storage.folders.create
- storage.objects.list
```

b. From Google Cloud, activate cloud shell.

c. Upload the YAML file that includes the required permissions.

d. Create a custom role by using the `gcloud iam roles create` command.

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

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

Google Cloud docs: Creating and managing custom roles

2. Assign this custom role to the user who will deploy the Console agent from the Console or by using gcloud.

Google Cloud docs: Grant a single role

Step 3: Create a Google Cloud service account to use with the agent

A Google Cloud service account is required to provide the Console agent with the permissions that the Console needs to manage resources in Google Cloud. When you create the Console agent, you'll need to associate this service account with the Console agent VM.

It's your responsibility to update the custom role as new permissions are added in subsequent releases. If new permissions are required, they will be listed in the release notes.

Steps

1. Create a custom role in Google Cloud:

a. Create a YAML file that includes the contents of the service account permissions for the Console agent.

b. From Google Cloud, activate cloud shell.

c. Upload the YAML file that includes the required permissions.

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 connector --project=myproject --file=agent.yaml
```

Google Cloud docs: Creating and managing custom roles

2. Create a service account in Google Cloud and assign the role to the service account:

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

3. If you plan to deploy Cloud Volumes ONTAP systems in different projects than the project where the Console agent resides, then you'll need to provide the Console agent's service account with access to those projects.

   For example, let's say the Console agent is in project 1 and you want to create Cloud Volumes ONTAP systems in project 2. You'll need to grant access to the service account in project 2.

   a. From the IAM & Admin service, select the Google Cloud project where you want to create Cloud Volumes ONTAP systems.
   b. On the **IAM** page, select **Grant Access** and provide the required details.
      - Enter the email of the Console agent's service account.
      - Select the Console agent's custom role.
      - Select **Save**.

   For more details, refer to Google Cloud documentation

Step 4: Set up shared VPC permissions

If you are using a shared VPC to deploy resources into a service project, then you'll need to prepare your permissions.

This table is for reference and your environment should reflect the permissions table when IAM configuration is complete.

View shared VPC permissions

Identity	Creator	Hosted in	Service project permissions	Host project permissions	Purpose
Google account to deploy the agent	Custom	Service Project	Agent deployment policy	compute.networkUser	Deploying the agent in the service project
agent service account	Custom	Service project	Agent service account policy	compute.networkUser deploymentmanager.editor	Deploying and maintaining Cloud Volumes ONTAP and services in the service project
Cloud Volumes ONTAP service account	Custom	Service project	storage.admin member: NetApp Console service account as serviceAccount.user	N/A	(Optional) For NetApp Cloud Tiering and NetApp Backup and Recovery
Google APIs service agent	Google Cloud	Service project	(Default) Editor	compute.networkUser	Interacts with Google Cloud APIs on behalf of deployment. Allows the Console to use the shared network.
Google Compute Engine default service account	Google Cloud	Service project	(Default) Editor	compute.networkUser	Deploys Google Cloud instances and compute infrastructure on behalf of deployment. Allows the Console to use the shared network.

Notes:

1. deploymentmanager.editor is only required at the host project if you are not passing firewall rules to the deployment and are choosing to let the Console create them for you. The NetApp Console creates a deployment in the host project which contains the VPC0 firewall rule if no rule is specified.
2. firewall.create and firewall.delete are only required if you are not passing firewall rules to the deployment and are choosing to let the Console create them for you. These permissions reside in the Console account .yaml file. If you are deploying an HA pair using a shared VPC, these permissions will be used to create the firewall rules for VPC1, 2 and 3. For all other deployments, these permissions will also be used to create rules for VPC0.
3. For Cloud Tiering, the tiering service account must have the serviceAccount.user role on the service account, not just at the project level. Currently if you assign serviceAccount.user at the project level, the permissions don't show when you query the service account with getIAMPolicy.

Step 5: Enable Google Cloud APIs

You must enable several Google Cloud APIs before deploying the Console agent and Cloud Volumes ONTAP.

Step

1. Enable the following Google Cloud APIs in your project:
    - Cloud Infrastructure Manager API
    - Cloud Deployment Manager V2 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))

Google Cloud documentation: Enabling APIs

Step 6: Create the Console agent

Create a Console agent directly from the Console.

Creating the Console agent deploys a virtual machine instance in Google Cloud using a default configuration. Do not switch to a smaller VM instance with fewer CPUs or less RAM after creating the Console agent. Learn about the default configuration for the Console agent.

When you deploy an agent in Google Cloud, the agent creates a bucket to store deployment files.

Before you begin

You should have the following:

- The required Google Cloud permissions to create the Console agent and a service account for the Console agent VM.
- A VPC and subnet that meets networking requirements.
- Details about a proxy server, if a proxy is required for internet access from the Console agent.

Steps

1. Select **Administration > Agents**.
2. On the **Overview** page, select **Deploy agent > Google Cloud**
3. On the **Deploying an agent** page, review the details about what you'll need. You have two options:
    a. Select **Continue** to prepare for deployment by using the in-product guide. Each step in the in-product guide includes the information that's contained on this page of the documentation.
    b. Select **Skip to Deployment** if you already prepared by following the steps on this page.
4. Follow the steps in the wizard to create the Console agent:
    - If you're prompted, log in to your Google account, which should have the required permissions to create the virtual machine instance.

      The form is owned and hosted by Google. Your credentials are not provided to NetApp.

- **Details**: Enter a name for the virtual machine instance, specify tags, select a project, and then select the service account that has the required permissions (refer to the section above for details).
- **Location**: Specify a region, zone, VPC, and subnet for the instance.
- **Network**: Choose whether to enable a public IP address and optionally specify a proxy configuration.
- **Network tags**: Add a network tag to the Console agent instance if using a transparent proxy. Network tags must start with a lowercase letter and can contain lowercase letters, numbers, and hyphens. Tags must end with a lowercase letter or number. For example, you might use the tag "console-agent-proxy".
- **Firewall Policy**: Choose whether to create a new firewall policy or whether to select an existing firewall policy that allows the required inbound and outbound rules.

  Firewall rules in Google Cloud

5. Review your selections to verify that your set up is correct.

   a. The **Validate agent configuration** check box is marked by default to have the Console validate the network connectivity requirements when you deploy. If the Console fails to deploy the agent, it provides a report to help you troubleshoot. If the deployment succeeds, no report is provided.

   If you are still using the previous endpoints used for agent upgrades, the validation fails with an error. To avoid this, unmark the check box to skip the validation check.

6. Select **Add**.

   The agent is ready in approximately 10 minutes; stay on the page until the process completes.

Result

After the process completes, the Console agent is available for use.

If the deployment fails, you can download a report and logs from the Console to help you fix the issues. Learn how to troubleshoot installation issues.

If you have Google Cloud Storage buckets in the same Google Cloud account where you created the Console agent, you'll see a Google Cloud Storage system appear on the **Systems** page automatically. Learn how to manage Google Cloud Storage from the Console

Create a Console agent from Google Cloud

To create a Console agent in Google Cloud by using Google Cloud, you need to set up your networking, prepare Google Cloud permissions, enable Google Cloud APIs, and then create the Console agent.

Before you begin

- You should have a understanding of Console agents.
- You should review Console agent limitations.

Step 1: Set up networking

Set up networking to enable the Console agent to manage resources and connect to target networks and the internet.

VPC and subnet

When you create the Console agent, you need to specify the VPC and subnet where it should reside.

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 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
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.
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.

Endpoints contacted from the NetApp console

As you use the web-based NetApp Console that's provided through the SaaS layer, it contacts several endpoints to complete data management tasks. This includes endpoints that are contacted to deploy the Console agent from the the Console.

View the list of endpoints contacted from the NetApp console.

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

Implement this networking requirement after creating the Console agent.

Step 2: Set up permissions to create the Console agent

Set up permissions for the Google Cloud user to deploy the Console agent VM from Google Cloud.

Steps

1. Create a custom role in Google Platform:
   a. Create a YAML file that includes the following permissions:

```
title: Console agent deployment policy
description: Permissions for the user who deploys the Console
agent
stage: GA
includedPermissions:

- cloudbuild.builds.get
- compute.disks.create
- compute.disks.get
- compute.disks.list
- compute.disks.setLabels
- compute.disks.use
- compute.firewalls.create
- compute.firewalls.delete
- compute.firewalls.get
- compute.firewalls.list
- compute.globalOperations.get
- compute.images.get
- compute.images.getFromFamily
- compute.images.list
- compute.images.useReadOnly
- compute.instances.attachDisk
- compute.instances.create
- compute.instances.get
- compute.instances.list
- compute.instances.setDeletionProtection
- compute.instances.setLabels
- compute.instances.setMachineType
- compute.instances.setMetadata
- compute.instances.setTags
- compute.instances.start
- compute.instances.updateDisplayDevice
- compute.machineTypes.get
- compute.networks.get
- compute.networks.list
- compute.networks.updatePolicy
- compute.projects.get
- compute.regions.get
- compute.regions.list
- compute.subnetworks.get
- compute.subnetworks.list
- compute.zoneOperations.get
- compute.zones.get
- compute.zones.list
- config.deployments.create
```

```
- config.operations.get
- config.deployments.delete
- config.deployments.deleteState
- config.deployments.get
- config.deployments.getState
- config.deployments.list
- config.deployments.update
- config.deployments.updateState
- config.previews.get
- config.previews.list
- config.revisions.get
- config.resources.list
- deploymentmanager.compositeTypes.get
- deploymentmanager.compositeTypes.list
- deploymentmanager.deployments.create
- deploymentmanager.deployments.delete
- deploymentmanager.deployments.get
- deploymentmanager.deployments.list
- deploymentmanager.manifests.get
- deploymentmanager.manifests.list
- deploymentmanager.operations.get
- deploymentmanager.operations.list
- deploymentmanager.resources.get
- deploymentmanager.resources.list
- deploymentmanager.typeProviders.get
- deploymentmanager.typeProviders.list
- deploymentmanager.types.get
- deploymentmanager.types.list
- resourcemanager.projects.get
- compute.instances.setServiceAccount
- iam.serviceAccounts.actAs
- iam.serviceAccounts.create
- iam.serviceAccounts.list
- iam.serviceAccountKeys.create
- storage.buckets.create
- storage.buckets.get
- storage.objects.create
- storage.folders.create
- storage.objects.list
```

b. From Google Cloud, activate cloud shell.

c. Upload the YAML file that includes the required permissions.

d. Create a custom role by using the `gcloud iam roles create` command.

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

gcloud iam roles create connectorDeployment --project=myproject --file=connector-deployment.yaml

Google Cloud docs: Creating and managing custom roles

2. Assign this custom role to the user who deploys the Console agent from Google Cloud.

Google Cloud docs: Grant a single role

Step 3: Set up permissions for the Console agent operations

A Google Cloud service account is required to provide the Console agent with the permissions that the Console needs to manage resources in Google Cloud. When you create the Console agent, you'll need to associate this service account with the Console agent VM.

It's your responsibility to update the custom role as new permissions are added in subsequent releases. If new permissions are required, they will be listed in the release notes.

Steps

1. Create a custom role in Google Cloud:
   a. Create a YAML file that includes the contents of the service account permissions for the Console agent.
   b. From Google Cloud, activate cloud shell.
   c. Upload the YAML file that includes the required permissions.
   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 connector --project=myproject --file=agent.yaml
      ```

      Google Cloud docs: Creating and managing custom roles

2. Create a service account in Google Cloud and assign the role to the service account:
   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

3. If you plan to deploy Cloud Volumes ONTAP systems in different projects than the project where the Console agent resides, then you'll need to provide the Console agent's service account with access to those projects.

   For example, let's say the Console agent is in project 1 and you want to create Cloud Volumes ONTAP systems in project 2. You'll need to grant access to the service account in project 2.

   a. From the IAM & Admin service, select the Google Cloud project where you want to create Cloud Volumes ONTAP systems.
   b. On the **IAM** page, select **Grant Access** and provide the required details.
      - Enter the email of the Console agent's service account.

- Select the Console agent's custom role.
- Select **Save**.

For more details, refer to Google Cloud documentation

Step 4: Set up shared VPC permissions

If you are using a shared VPC to deploy resources into a service project, then you'll need to prepare your permissions.

This table is for reference and your environment should reflect the permissions table when IAM configuration is complete.

View shared VPC permissions

Identity	Creator	Hosted in	Service project permissions	Host project permissions	Purpose
Google account to deploy the agent	Custom	Service Project	Agent deployment policy	compute.networkUser	Deploying the agent in the service project
agent service account	Custom	Service project	Agent service account policy	compute.networkUser deploymentmanager.editor	Deploying and maintaining Cloud Volumes ONTAP and services in the service project
Cloud Volumes ONTAP service account	Custom	Service project	storage.admin member: NetApp Console service account as serviceAccount.user	N/A	(Optional) For NetApp Cloud Tiering and NetApp Backup and Recovery
Google APIs service agent	Google Cloud	Service project	(Default) Editor	compute.networkUser	Interacts with Google Cloud APIs on behalf of deployment. Allows the Console to use the shared network.
Google Compute Engine default service account	Google Cloud	Service project	(Default) Editor	compute.networkUser	Deploys Google Cloud instances and compute infrastructure on behalf of deployment. Allows the Console to use the shared network.

Notes:

1. deploymentmanager.editor is only required at the host project if you are not passing firewall rules to the deployment and are choosing to let the Console create them for you. The NetApp Console creates a deployment in the host project which contains the VPC0 firewall rule if no rule is specified.
2. firewall.create and firewall.delete are only required if you are not passing firewall rules to the deployment and are choosing to let the Console create them for you. These permissions reside in the Console account .yaml file. If you are deploying an HA pair using a shared VPC, these permissions will be used to create the firewall rules for VPC1, 2 and 3. For all other deployments, these permissions will also be used to create rules for VPC0.
3. For Cloud Tiering, the tiering service account must have the serviceAccount.user role on the service account, not just at the project level. Currently if you assign serviceAccount.user at the project level, the permissions don't show when you query the service account with getIAMPolicy.

Step 5: Enable Google Cloud APIs

Enable several Google Cloud APIs before deploying the Console agent and Cloud Volumes ONTAP.

Step

1. Enable the following Google Cloud APIs in your project:
   - Cloud Infrastructure Manager API
   - Cloud Deployment Manager V2 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))

Google Cloud documentation: Enabling APIs

Step 6: Create the Console agent

Create a Console agent by using Google Cloud.

Creating the Console agent deploys a VM instance in Google Cloud with the default configuration. Do not switch to a smaller VM instance with fewer CPUs or less RAM after you create the Console agent. Learn about the default configuration for the Console agent.

Before you begin

You should have the following:

- The required Google Cloud permissions to create the Console agent and a service account for the Console agent VM.
- A VPC and subnet that meets networking requirements.
- An understanding of VM instance requirements.
  - **CPU**: 8 cores or 8 vCPUs
  - **RAM**: 32 GB
  - **Machine type**: We recommend n2-standard-8.

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

Steps

1. Log in to the Google Cloud SDK using your preferred method.

   This example uses a local shell with the gcloud SDK installed, but you can also use the Google Cloud Shell.

   For more information about the Google Cloud SDK, visit the Google Cloud SDK documentation page.

2. Verify that you are logged in as a user who has the required permissions that are defined in the section above:

```
gcloud auth list
```

The output should show the following where the * user account is the desired user account to be logged in as:

```
Credentialed Accounts
ACTIVE  ACCOUNT
     some_user_account@domain.com
*    desired_user_account@domain.com
To set the active account, run:
 $ gcloud config set account `ACCOUNT`
Updates are available for some Cloud SDK components. To install them,
please run:
$ gcloud components update
```

3. Run the `gcloud compute instances create` command:

```
gcloud compute instances create <instance-name>
  --machine-type=n2-standard-8
  --image-project=netapp-cloudmanager
  --image-family=cloudmanager
  --scopes=cloud-platform
  --project=<project>
  --service-account=<service-account>
  --zone=<zone>
  --no-address
  --tags <network-tag>
  --network <network-path>
  --subnet <subnet-path>
  --boot-disk-kms-key <kms-key-path>
```

instance-name
: The desired instance name for the VM instance.

project
: (Optional) The project where you want to deploy the VM.

service-account
: The service account specified in the output from step 2.

zone
: The zone where you want to deploy the VM

no-address

(Optional) No external IP address is used (you need a cloud NAT or proxy to route traffic to the public internet)

network-tag

(Optional) Add network tagging to link a firewall rule using tags to the Console agent instance

network-path

(Optional) Add the name of the network to deploy the Console agent into (for a Shared VPC, you need the full path)

subnet-path

(Optional) Add the name of the subnet to deploy the Console agent into (for a Shared VPC, you need the full path)

kms-key-path

(Optional) Add a KMS key to encrypt the Console agent's disks (IAM permissions also need to be applied)

For more information about these flags, visit the Google Cloud compute SDK documentation.

Running the command deploys the Console agent. The Console agent instance and software should be running in approximately five minutes.

4. 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.

5. After you log in, set up the Console agent:
   a. Specify the Console organization to associate with the Console agent.

      Learn about identity and access management.

   b. Enter a name for the system.

Result

The Console agent is now installed and set up with your Console organization.

Open a web browser and go to the NetApp Console to start using the Console agent.

Manually install the Console agent in Google Cloud

To manually install the Console agent on your own Linux host, you need to review host requirements, set up your networking, prepare Google Cloud permissions, enable Google Cloud APIs, install the Console, and then provide the permissions that you prepared.

Before you begin

- You should have an understanding of Console agents.
- You should review Console agent limitations.

Step 1: Review host requirements

The Console agent software must run on a host that meets specific operating system requirements, RAM requirements, port requirements, and so on.

> 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.

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

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				

Operating system	Supported OS versions	Supported agent versions	Required container tool	SELinux
	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
	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

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

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

Step 2: 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 3. 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 3: Set up networking

Set up your networking so the Console agent can manage resources and processes within your hybrid cloud environment. For example, you need to ensure that connections are available to target networks and that outbound internet access is available.

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.

Endpoints	Purpose
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.
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.

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

Step 4: Set up permissions for the Console agent

A Google Cloud service account is required to provide the Console agent with the permissions that the

Console needs to manage resources in Google Cloud. When you create the Console agent, you'll need to associate this service account with the Console agent VM.

It's your responsibility to update the custom role as new permissions are added in subsequent releases. If new permissions are required, they will be listed in the release notes.

Steps

1. Create a custom role in Google Cloud:
   a. Create a YAML file that includes the contents of the service account permissions for the Console agent.
   b. From Google Cloud, activate cloud shell.
   c. Upload the YAML file that includes the required permissions.
   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 connector --project=myproject --file=agent.yaml
      ```

      Google Cloud docs: Creating and managing custom roles

2. Create a service account in Google Cloud and assign the role to the service account:
   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

3. If you plan to deploy Cloud Volumes ONTAP systems in different projects than the project where the Console agent resides, then you'll need to provide the Console agent's service account with access to those projects.

   For example, let's say the Console agent is in project 1 and you want to create Cloud Volumes ONTAP systems in project 2. You'll need to grant access to the service account in project 2.

   a. From the IAM & Admin service, select the Google Cloud project where you want to create Cloud Volumes ONTAP systems.
   b. On the **IAM** page, select **Grant Access** and provide the required details.
      - Enter the email of the Console agent's service account.
      - Select the Console agent's custom role.
      - Select **Save**.

   For more details, refer to Google Cloud documentation

Step 5: Set up shared VPC permissions

If you are using a shared VPC to deploy resources into a service project, then you'll need to prepare your permissions.

This table is for reference and your environment should reflect the permissions table when IAM configuration is

complete.

View shared VPC permissions

Identity	Creator	Hosted in	Service project permissions	Host project permissions	Purpose
Google account to deploy the agent	Custom	Service Project	Agent deployment policy	compute.networkUser	Deploying the agent in the service project
agent service account	Custom	Service project	Agent service account policy	compute.networkUser deploymentmanager.editor	Deploying and maintaining Cloud Volumes ONTAP and services in the service project
Cloud Volumes ONTAP service account	Custom	Service project	storage.admin member: NetApp Console service account as serviceAccount.user	N/A	(Optional) For NetApp Cloud Tiering and NetApp Backup and Recovery
Google APIs service agent	Google Cloud	Service project	(Default) Editor	compute.networkUser	Interacts with Google Cloud APIs on behalf of deployment. Allows the Console to use the shared network.
Google Compute Engine default service account	Google Cloud	Service project	(Default) Editor	compute.networkUser	Deploys Google Cloud instances and compute infrastructure on behalf of deployment. Allows the Console to use the shared network.

Notes:

1. deploymentmanager.editor is only required at the host project if you are not passing firewall rules to the deployment and are choosing to let the Console create them for you. The NetApp Console creates a deployment in the host project which contains the VPC0 firewall rule if no rule is specified.
2. firewall.create and firewall.delete are only required if you are not passing firewall rules to the deployment and are choosing to let the Console create them for you. These permissions reside in the Console account .yaml file. If you are deploying an HA pair using a shared VPC, these permissions will be used to create the firewall rules for VPC1, 2 and 3. For all other deployments, these permissions will also be used to create rules for VPC0.
3. For Cloud Tiering, the tiering service account must have the serviceAccount.user role on the service account, not just at the project level. Currently if you assign serviceAccount.user at the project level, the permissions don’t show when you query the service account with getIAMPolicy.

Step 6: Enable Google Cloud APIs

Several Google Cloud APIs must be enabled before you can deploy a Console agent in Google Cloud.

Step

1. Enable the following Google Cloud APIs in your project:
    - Cloud Infrastructure Manager API
    - Cloud Deployment Manager V2 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))

Google Cloud documentation: Enabling APIs

Step 7: Install the Console agent

After the pre-requisites are complete, you can manually install the software on your own Linux host.

When you deploy an agent, the system also creates a Google Cloud bucket to store deployment files.

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.
2. Wait for the installation to complete.

   At the end of the installation, the Console agent service (occm) restarts twice if you specified a proxy server.

If the installation fails, you can view the installation report and logs to help you fix the issues. Learn how to troubleshoot installation issues.

1. Open a web browser from a host that has a connection to the Console agent virtual machine and enter the following URL:

   https://*ipaddress*

2. After you log in, set up the Console agent:

   a. Specify the 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.

      You should keep restricted mode disabled because these steps describe how to use the Console in standard mode. You should enable restricted mode only if you have a secure environment and want to disconnect this account from backend services. If that's the case, follow steps to get started with the NetApp Console in restricted mode.

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

   If the installation fails, you can view logs and a report to help you troubleshoot. Learn how to troubleshoot installation issues.

If you have Google Cloud Storage buckets in the same Google Cloud account where you created the Console agent, you'll see a Google Cloud Storage system appear on the **Systems** page automatically. Learn how to manage Google Cloud Storage from the NetApp Console

Step 8: Provide permissions to Console agent

You need to provide the Console agent with the Google Cloud permissions that you previously set up. Providing the permissions enables the Console agent to manage your data and storage infrastructure in Google Cloud.

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 Google Cloud 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.

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
	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

Operating system	Supported OS versions	Supported agent versions	Required container tool	SELinux
	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
	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.

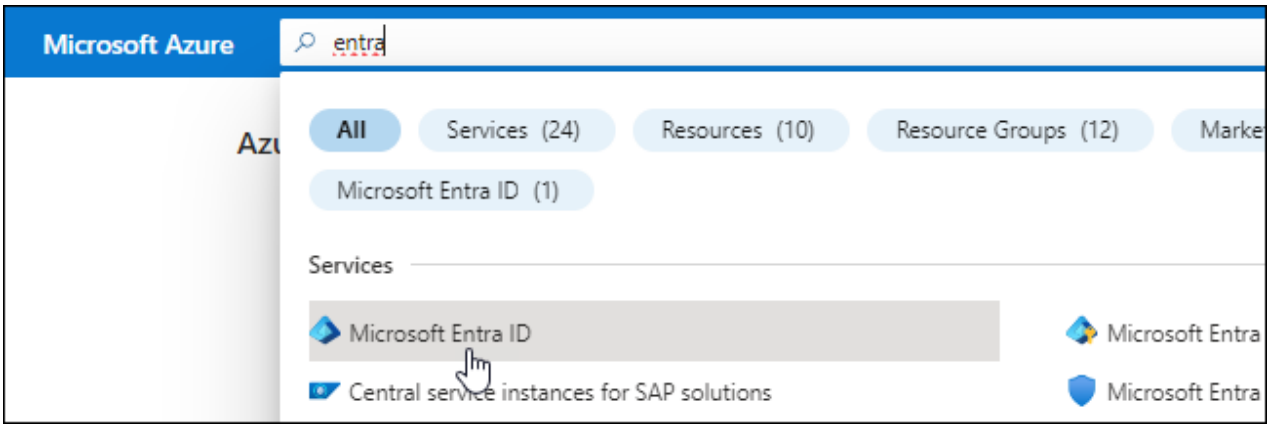
Azure

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.

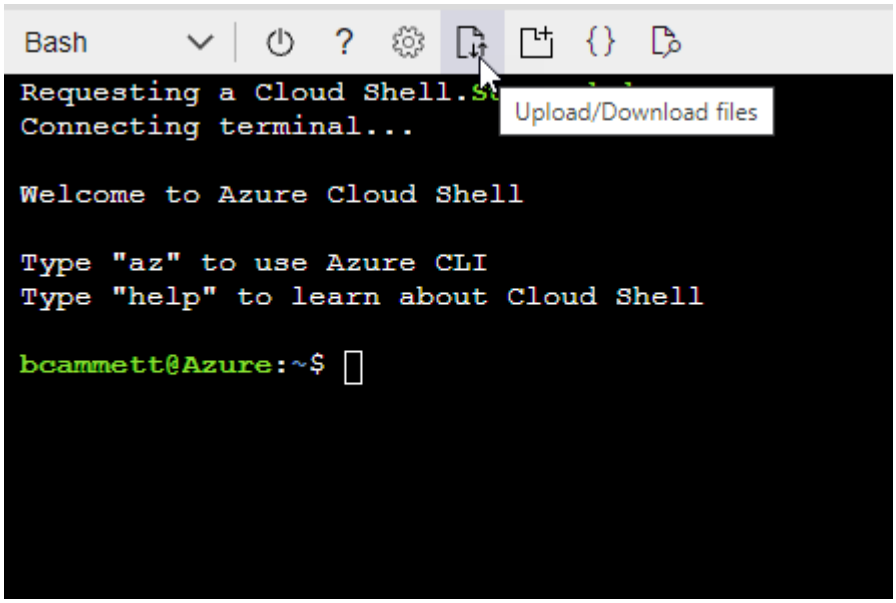
      Example

```
"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.

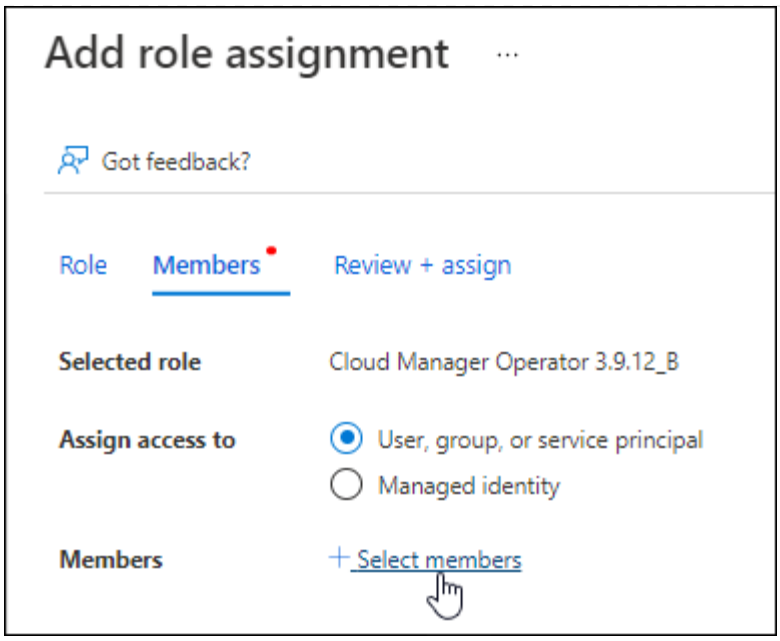


- Use the Azure CLI to create the custom role:

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

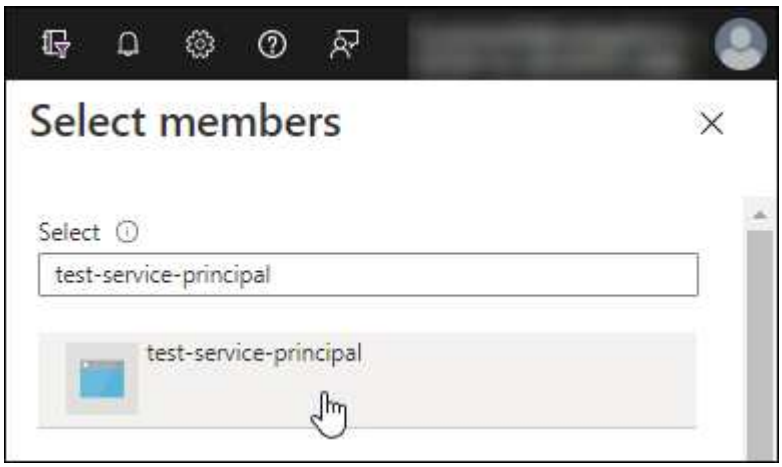
You should now have a custom role called Console Operator that you can assign to the Console agent virtual machine.

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**.

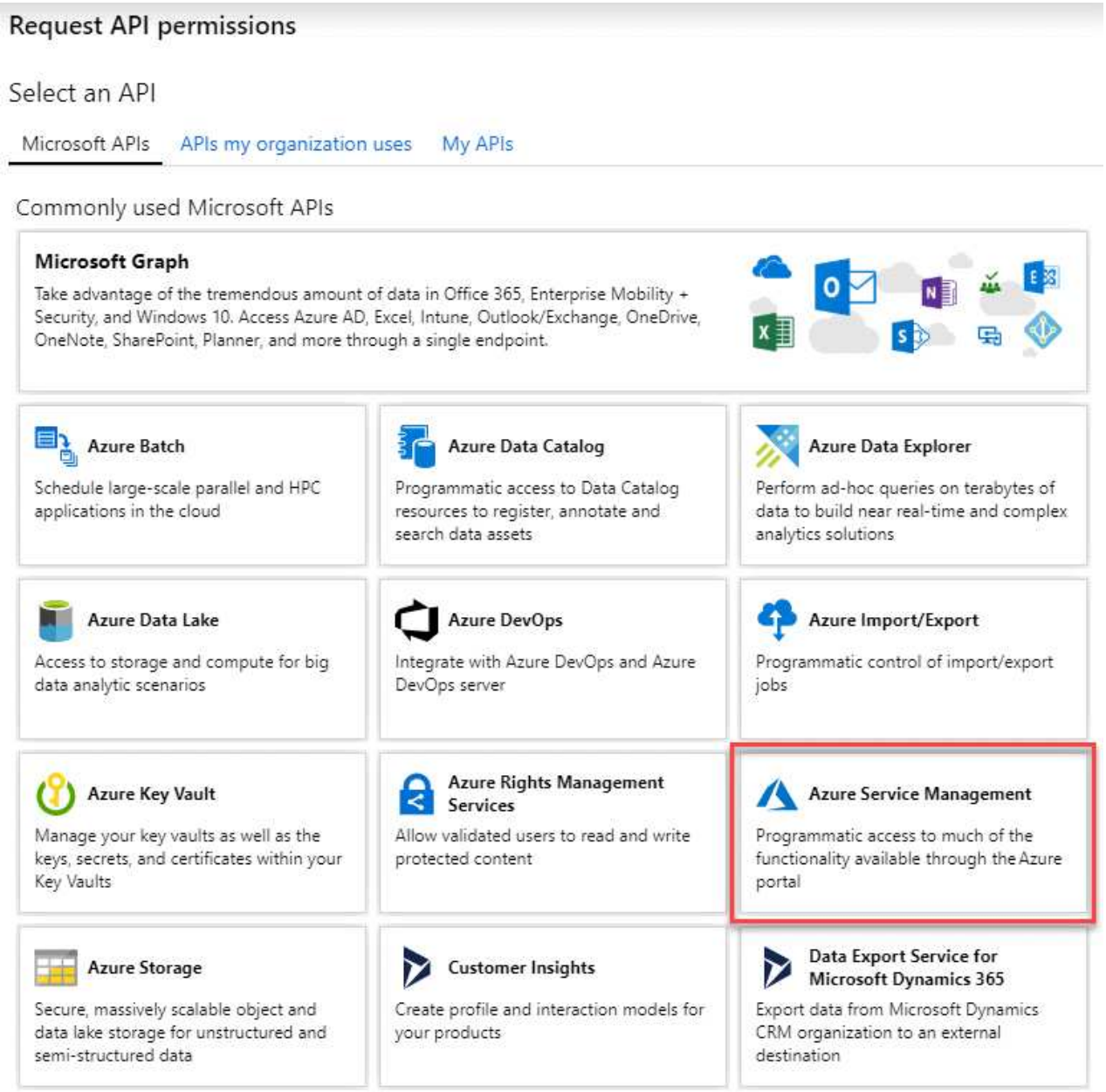
f. Select **Review + assign**.

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

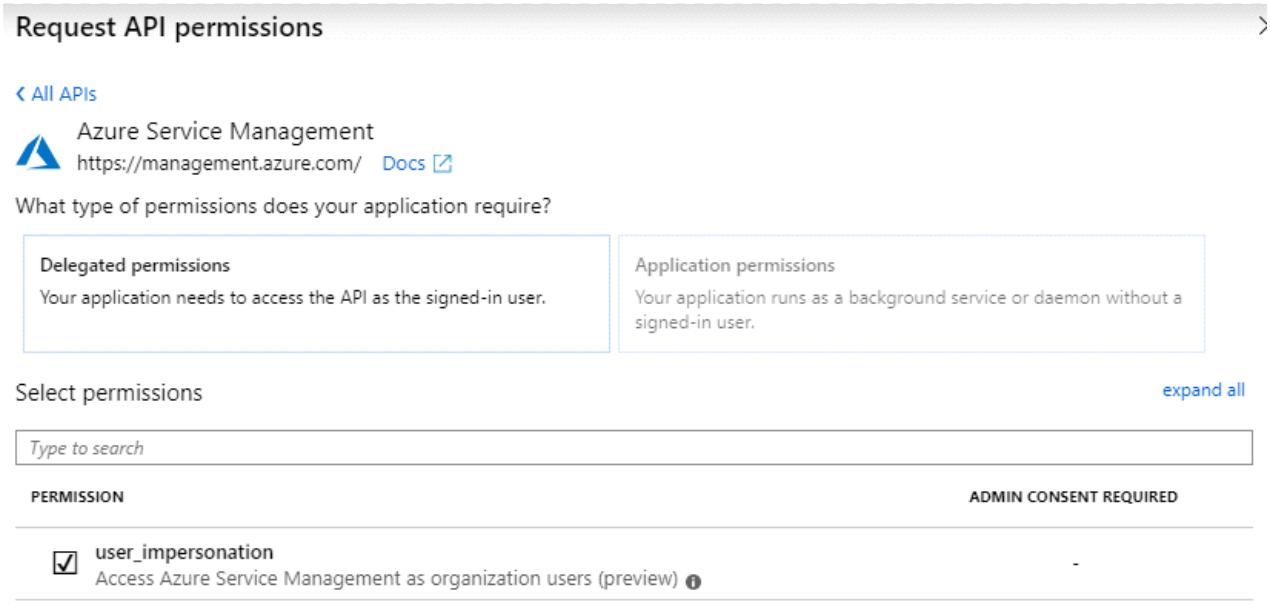
If you want to deploy Cloud Volumes ONTAP from multiple Azure subscriptions, then you must bind the service principal to each of those subscriptions. In the NetApp Console, you can select the subscription that you want to use when deploying Cloud Volumes ONTAP.

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**.

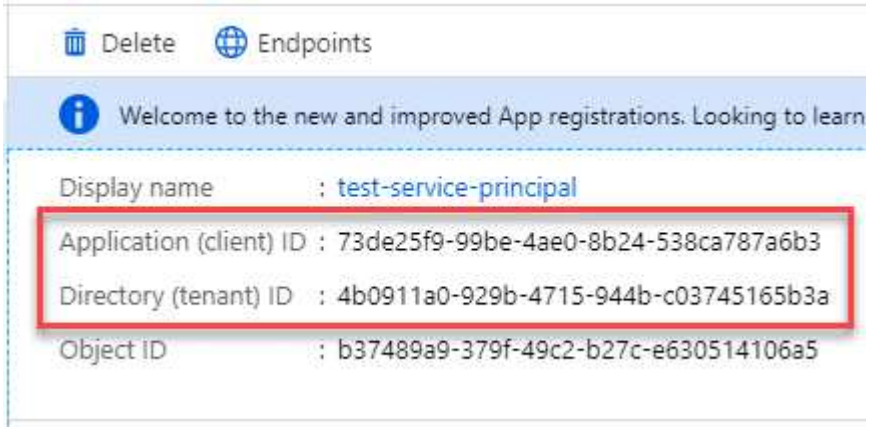


4. Select **Access Azure Service Management as organization users** and then select **Add permissions**.



Get the application ID and directory ID for the application

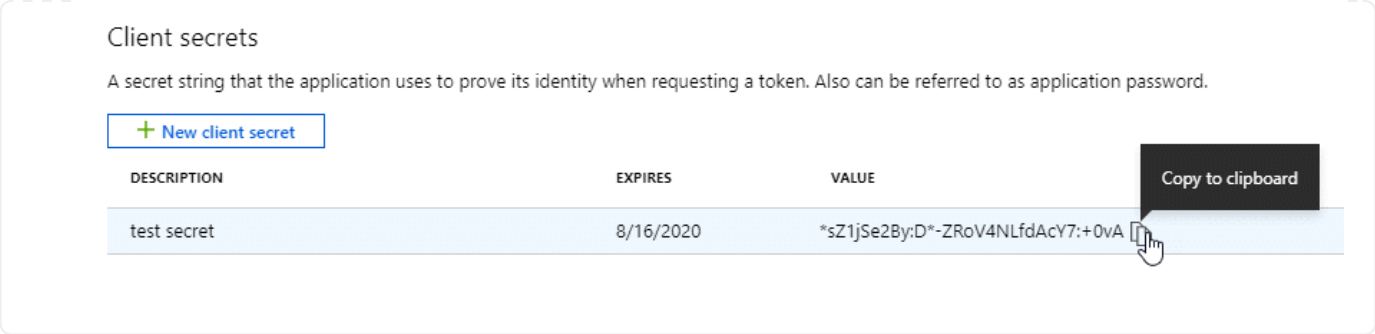
1. In the **Microsoft Entra ID** service, select **App registrations** and select the application.
2. Copy the **Application (client) ID** and the **Directory (tenant) ID**.



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 4. 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.

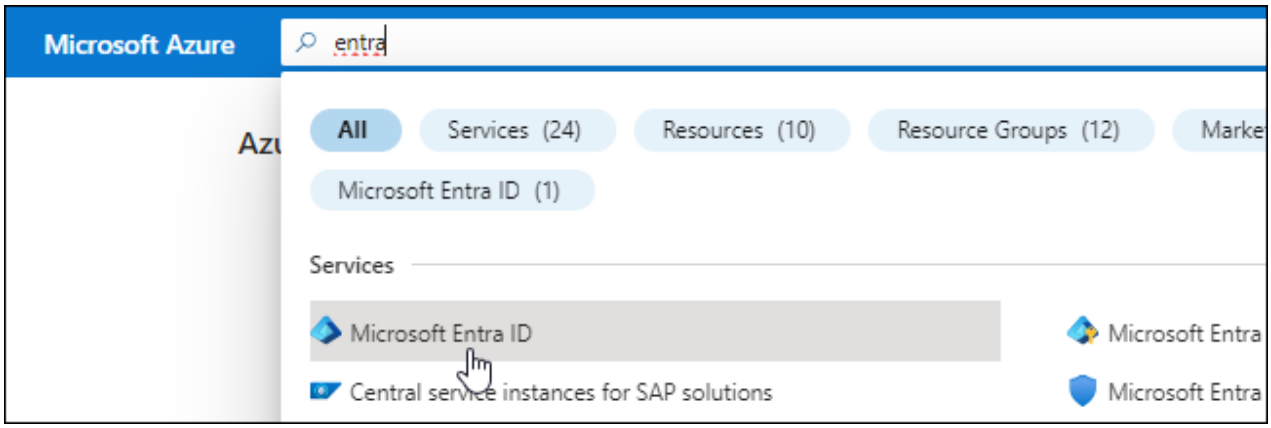
Azure

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.

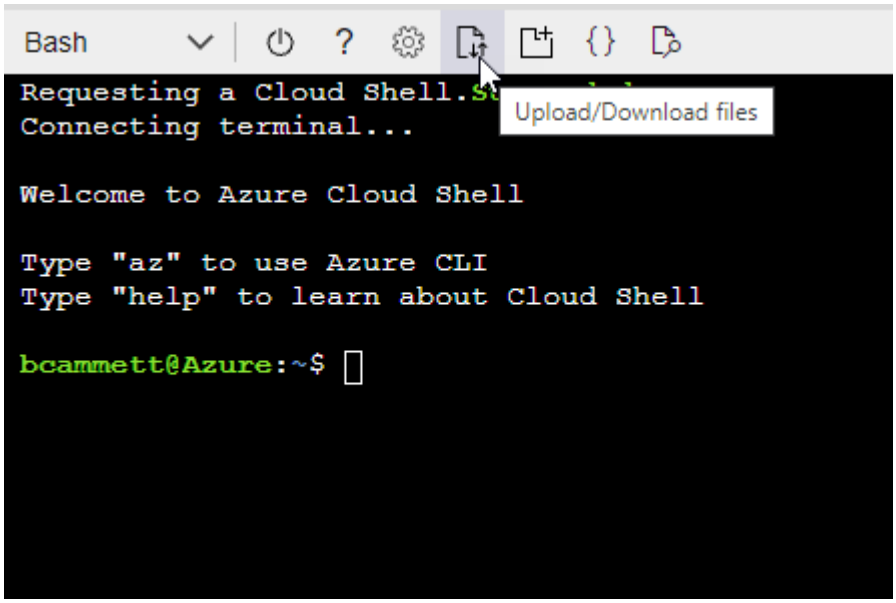
      Example

```
"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.

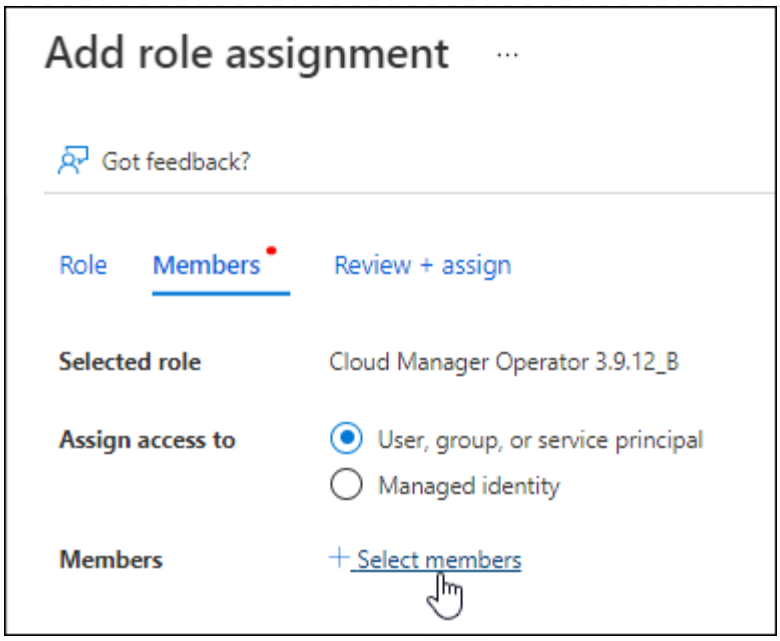


- Use the Azure CLI to create the custom role:

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

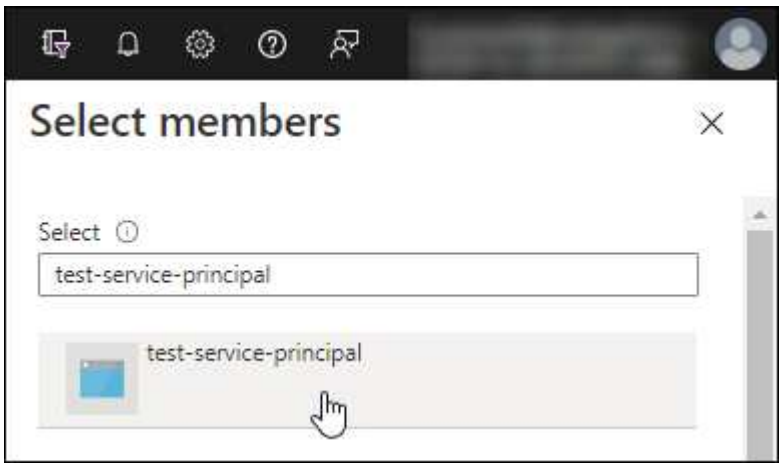
You should now have a custom role called Console Operator that you can assign to the Console agent virtual machine.

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**.

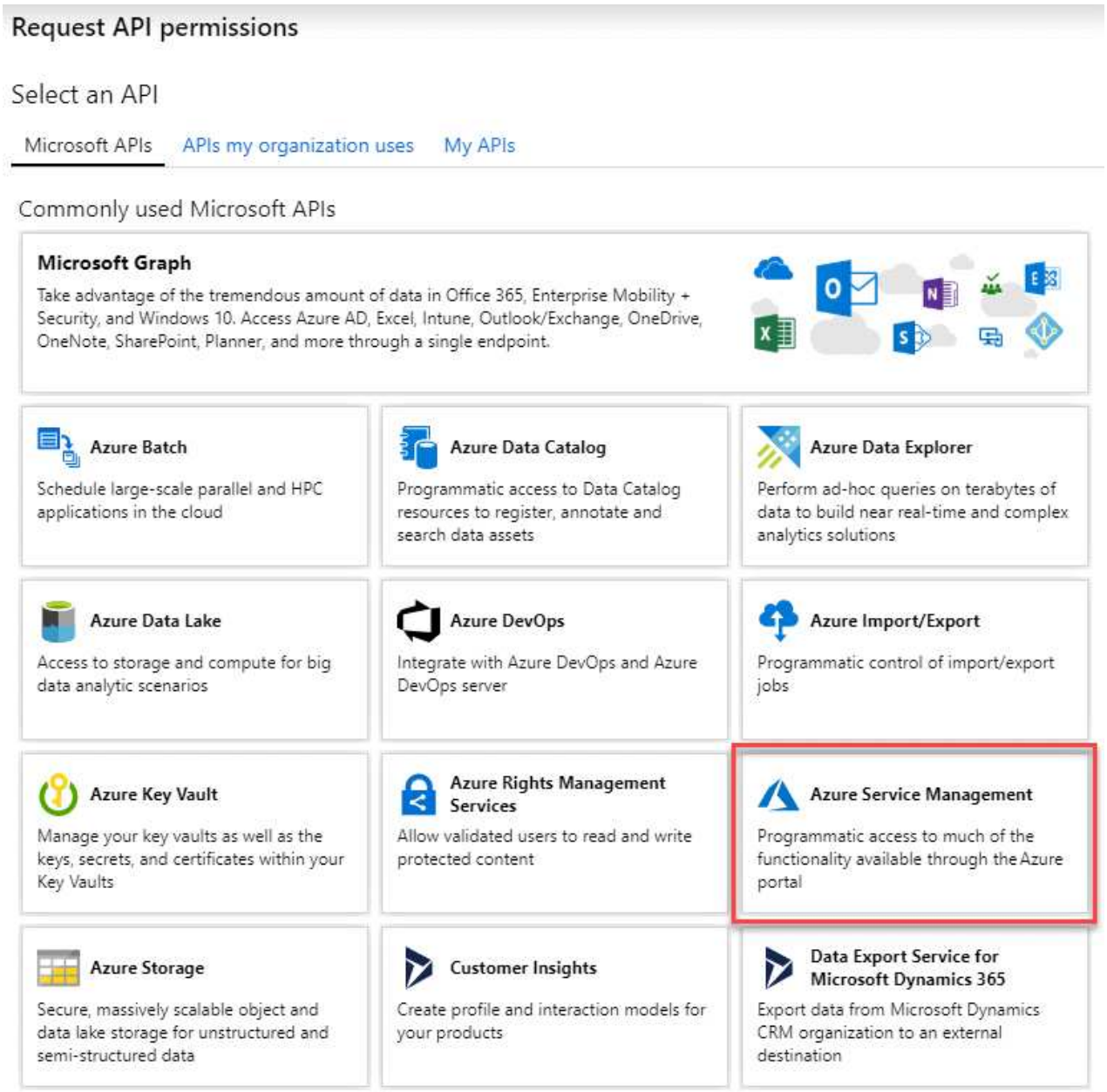
f. Select **Review + assign**.

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

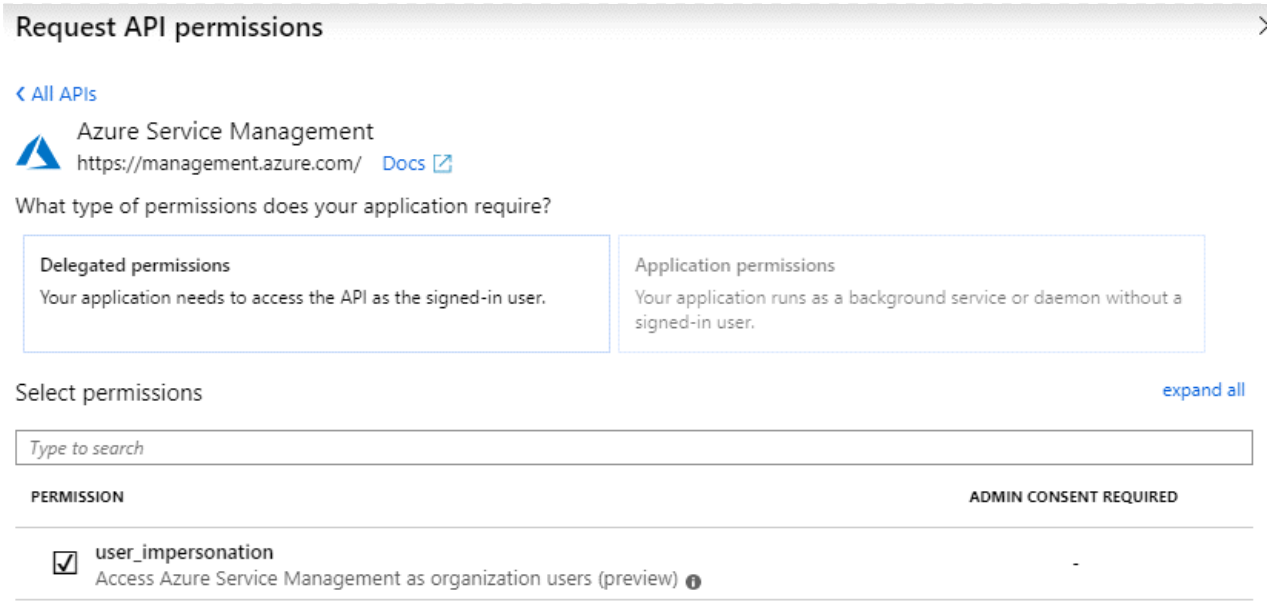
If you want to deploy Cloud Volumes ONTAP from multiple Azure subscriptions, then you must bind the service principal to each of those subscriptions. In the NetApp Console, you can select the subscription that you want to use when deploying Cloud Volumes ONTAP.

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**.

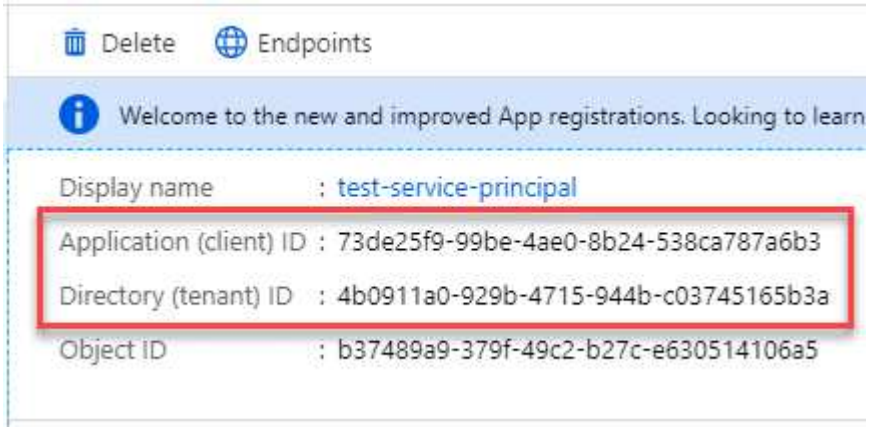


4. Select **Access Azure Service Management as organization users** and then select **Add permissions**.



Get the application ID and directory ID for the application

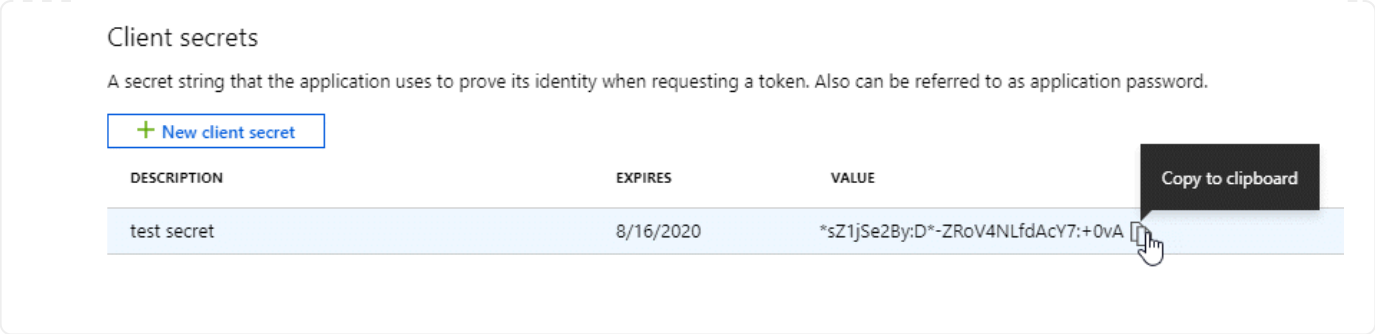
1. In the **Microsoft Entra ID** service, select **App registrations** and select the application.
2. Copy the **Application (client) ID** and the **Directory (tenant) ID**.



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