



Set up a Connector

Set up and administration

NetApp

January 17, 2023

Table of Contents

- Set up a Connector 1
 - Learn about Connectors..... 1
 - Create a Connector in AWS from BlueXP 5
 - Create a Connector in Azure from BlueXP..... 11
 - Create a Connector in Google Cloud from BlueXP 27
 - Create a Connector in a Government region 39

Set up a Connector

Learn about Connectors

In most cases, a BlueXP Account Admin will need to deploy a *Connector* in your cloud or on-premises network. The Connector is a crucial component for the day-to-day use of BlueXP. It enables BlueXP to manage the resources and processes within your public cloud environment.

When a Connector is required

A Connector is required for the following features and services in BlueXP:

- Amazon FSx for ONTAP management features
- Amazon S3 discovery
- Azure Blob discovery
- Cloud Backup
- Cloud Data Sense
- Cloud Tiering
- Cloud Volumes ONTAP
- E-Series systems
- Global File Cache
- Google Cloud Storage discovery
- Kubernetes clusters
- On-premises ONTAP cluster integration with BlueXP data services
- StorageGRID

A Connector is **not** required for the following services:

- Digital Advisor

In almost all cases, you can add a license to the Digital Wallet without a Connector.

The only time that a Connector is required to add a license to the Digital Wallet is for Cloud Volumes ONTAP *node-based* licenses. A Connector is required in this case because the data is taken from the licenses installed on Cloud Volumes ONTAP systems.

- Amazon FSx for ONTAP working environment creation

While a Connector isn't required to create a working environment, it is required to create and manage volumes, replicate data, and integrate FSx for ONTAP with NetApp cloud services, such as Data Sense and Cloud Sync.

- Azure NetApp Files

While a Connector isn't required to set up and manage Azure NetApp Files, a Connector is required if you want to use Cloud Data Sense to scan Azure NetApp Files data.

- Cloud Volumes Service for Google Cloud
- Cloud Sync
- Direct discovery of on-premises ONTAP clusters

While a Connector isn't required for direct discovery of an on-premises ONTAP cluster, a Connector is required if you want to take advantage of additional BlueXP features.

[Learn more about discovery and management options for on-prem ONTAP clusters](#)

Supported locations

A Connector is supported in the following locations:

- Amazon Web Services
- Microsoft Azure
- Google Cloud
- On your premises
- On your premises, without internet access

Note about Azure deployments

If you deploy the Connector in Azure, it 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.](#)

Note about Google Cloud deployments

If you want to create a Cloud Volumes ONTAP system in Google Cloud, then you must have a Connector that's running in Google Cloud as well. You can't use a Connector that's running in AWS, Azure, or on-prem.

Connectors should remain running

A Connector should remain running at all times. It's important for the continued health and operation of the services that you enable.

For example, a Connector is a key component in the health and operation of Cloud Volumes ONTAP. If a Connector is powered down, Cloud Volumes ONTAP PAYGO systems and capacity-based BYOL systems shut down after losing communication with a Connector for longer than 14 days. This happens because the Connector refreshes licensing on the system each day.



If your Cloud Volumes ONTAP system has a node-based BYOL license, the system remains running after 14 days because the license is installed on the Cloud Volumes ONTAP system.

How to create a Connector

A BlueXP Account Admin can create a Connector in a number of ways:

- Directly from BlueXP (recommended)

- [Create in AWS](#)
- [Create in Azure](#)
- [Create in GCP](#)
- By manually installing the software on your own Linux host
 - [On a host that has internet access](#)
 - [On an on-prem host that doesn't have internet access](#)
- From your cloud provider's marketplace
 - [AWS Marketplace](#)
 - [Azure Marketplace](#)

If you are operating in a Government region, you need to deploy a Connector from your cloud provider's marketplace or by manually installing the Connector software on an existing Linux host. You can't deploy the Connector in a Government region from BlueXP's SaaS website.

Permissions

Specific permissions are needed to create the Connector and another set of permissions are needed for the Connector instance itself.

Permissions to create a Connector

The user who creates a Connector from BlueXP needs specific permissions to deploy the instance in your cloud provider of choice.

- [View the required AWS permissions](#)
- [View the required Azure permissions](#)
- [View the required Google Cloud permissions](#)

Permissions for the Connector instance

The Connector needs specific cloud provider permissions to perform operations on your behalf. For example, to deploy and manage Cloud Volumes ONTAP.

When you create a Connector directly from BlueXP, BlueXP creates the Connector with the permissions that it needs. There's nothing that you need to do.

If you create the Connector yourself from the AWS Marketplace, the Azure Marketplace, or by manually installing the software, then you'll need to make sure that the right permissions are in place.

- [Learn how the Connector uses AWS permissions](#)
- [Learn how the Connector uses Azure permissions](#)
- [Learn how the Connector uses Google Cloud permissions](#)

Connector upgrades

We typically update the Connector software each month to introduce new features and to provide stability improvements. While most of the services and features in the BlueXP platform are offered through SaaS-based software, a few features and functionalities are dependent on the version of the Connector. That includes Cloud Volumes ONTAP management, on-prem ONTAP cluster management, settings, and help.

The Connector automatically updates its software to the latest version, as long as it has outbound internet access to obtain the software update.

Number of working environments per Connector

A Connector can manage multiple working environments in BlueXP. The maximum number of working environments that a single Connector should manage varies. It depends on the type of working environments, the number of volumes, the amount of capacity being managed, and the number of users.

If you have a large-scale deployment, work with your NetApp representative to size your environment. If you experience any issues along the way, reach out to us by using the in-product chat.

When to use multiple Connectors

In some cases, you might only need one Connector, but you might find yourself needing two or more Connectors.

Here are a few examples:

- You're using a multi-cloud environment (AWS and Azure), so you have one Connector in AWS and another in Azure. Each manages the Cloud Volumes ONTAP systems running in those environments.
- A service provider might use one NetApp account to provide services for their customers, while using another account to provide disaster recovery for one of their business units. Each account would have separate Connectors.

Using multiple Connectors with the same working environment

You can manage a working environment with multiple Connectors at the same time for disaster recovery purposes. If one Connector goes down, you can switch to the other Connector to immediately manage the working environment.

To set up this configuration:

1. [Switch to another Connector](#)
2. Discover the existing working environment.
 - [Add existing Cloud Volumes ONTAP systems to BlueXP](#)
 - [Discover ONTAP clusters](#)
3. Set the [Capacity Management Mode](#)

Only the main Connector should be set to **Automatic Mode**. If you switch to another Connector for DR purposes, then you can change the Capacity Management Mode as needed.

When to switch between Connectors

When you create your first Connector, BlueXP automatically uses that Connector for each additional working environment that you create. Once you create an additional Connector, you'll need to switch between them to see the working environments that are specific to each Connector.

[Learn how to switch between Connectors.](#)

The local user interface

While you should perform almost all tasks from the [SaaS user interface](#), a local user interface is still available on the Connector. This interface is needed if you install the Connector in an environment that doesn't have internet access (like a Government region), and for a few tasks that need to be performed from the Connector itself, instead of the SaaS interface:

- [Setting a proxy server](#)
- Installing a patch (you'll typically work with NetApp personnel to install a patch)
- Downloading AutoSupport messages (usually directed by NetApp personnel when you have issues)

[Learn how to access the local UI.](#)

Create a Connector in AWS from BlueXP

A BlueXP Account Admin needs to deploy a *Connector* before you can use most BlueXP features. The Connector enables BlueXP to manage resources and processes within your public cloud environment.

These steps describe how to create a Connector in an AWS commercial region directly from the BlueXP SaaS website.

- [Learn how to deploy a Connector in a Government region](#)
- [Learn about other ways to deploy a Connector](#)

Quick start

Get started quickly by following these steps, or scroll down to the remaining sections for full details.

1

Set up authentication

To launch the Connector in AWS, BlueXP needs to authenticate with AWS by either assuming an IAM role or by using AWS access keys. With either option, an IAM policy is required.

[View the IAM role](#) or [follow step-by-step instructions](#).

2

Set up networking

You'll need a VPC and subnet with outbound internet access to specific endpoints. If an HTTP proxy is required for outbound internet, then you'll need the IP address, credentials, and HTTPS certificate.

[View networking requirements](#).

3

Create the Connector

Click the Connector drop-down, select **Add Connector** and follow the prompts.

[Follow step-by-step instructions](#).

Set up AWS authentication

BlueXP needs to authenticate with AWS before it can deploy the Connector instance in your VPC. You can choose one of these authentication methods:

- Let BlueXP 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, you first need to start by creating an IAM policy that includes the required permissions.

Create an IAM policy

This policy contains only the permissions needed to launch the Connector instance in AWS from BlueXP. Don't use this policy for other situations.

When BlueXP creates the Connector, it applies a new set of permissions to the Connector instance that enables the Connector to manage the resources in your public cloud environment.

Steps

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

```
{
  "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",
      "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",
        "cloudformation:CreateStack",
        "cloudformation:DeleteStack",
        "cloudformation:DescribeStacks",
        "cloudformation:DescribeStackEvents",
        "cloudformation:ValidateTemplate",
        "ec2:AssociateIamInstanceProfile",
        "ec2:DescribeIamInstanceProfileAssociations",
        "ec2:DisassociateIamInstanceProfile",
        "iam:GetRole",
        "iam:TagRole",
        "iam:ListRoles",
        "kms:ListAliases"
    ],
    "Resource": "*"
},
{
    "Effect": "Allow",
    "Action": [
        "ec2:TerminateInstances"
    ],
    "Condition": {
        "StringLike": {
            "ec2:ResourceTag/OCCMInstance": "*"
        }
    },
    "Resource": [
        "arn:aws:ec2:*:*:instance/*"
    ]
}
]
}

```

5. Click **Next** and add tags, if needed.

6. Click **Next** and enter a name and description.
7. Click **Create policy**.

What's next?

Either attach the policy to an IAM role that BlueXP can assume or to an IAM user.

Set up an IAM role

Set up an IAM role that BlueXP can assume in order to deploy the Connector in AWS.

Steps

1. Go to the AWS IAM console in the target account.
2. Under Access Management, click **Roles > Create Role** and follow the steps to create the role.

Be sure to do the following:

- Under **Trusted entity type**, select **AWS account**.
 - Select **Another AWS account** and enter the ID of the BlueXP SaaS account: 952013314444
 - Select the policy that you created in the previous section.
3. After you create the role, copy the Role ARN so that you can paste it in BlueXP when you create the Connector.

Result

The IAM role now has the required permissions.

Set up permissions for an IAM user

When you create a Connector, you can provide an AWS access key and secret key for an IAM user who has the required permissions to deploy the Connector instance.

Steps

1. From the AWS IAM console, click **Users** and then select the user name.
2. Click **Add permissions > Attach existing policies directly**.
3. Select the policy that you created.
4. Click **Next** and then click **Add permissions**.
5. Ensure that you have access to an access key and secret key for the IAM user.

Result

The AWS user now has the permissions required to create the Connector from BlueXP. You'll need to specify AWS access keys for this user when you're prompted by BlueXP.

Set up networking

Set up your networking so the Connector can manage resources and processes within your public cloud environment. Other than having a virtual network and subnet for the Connector, you'll need to ensure that the following requirements are met.


Connection to target networks

A Connector requires a network connection to the type of working environment that you're creating and the services that you're planning to enable.

For example, if you install a Connector in your corporate network, then you must set up a VPN connection to the virtual network in which you launch Cloud Volumes ONTAP.

Outbound internet access

The Connector requires outbound internet access to manage resources and processes within your public cloud environment.

Endpoints	Purpose
https://support.netapp.com	To obtain licensing information and to send AutoSupport messages to NetApp support.
https://*.api.bluexp.netapp.com https://api.bluexp.netapp.com https://*.cloudmanager.cloud.netapp.com https://cloudmanager.cloud.netapp.com	To provide SaaS features and services within BlueXP.  The Connector is currently contacting "cloudmanager.cloud.netapp.com" but it will start contacting "api.bluexp.netapp.com" in an upcoming release.
https://cloudmanagerinfraprod.azurecr.io https://*.blob.core.windows.net	To upgrade the Connector and its Docker components.

Proxy server

If your organization requires deployment of an HTTP proxy for all outgoing internet traffic, obtain the following information about your HTTP proxy:

- IP address
- Credentials
- HTTPS certificate

Security group

There's no incoming traffic to the Connector, unless you initiate it or if the Connector is used as a proxy for AutoSupport messages. HTTP and HTTPS provide access to the [local UI](#), which you'll use in rare circumstances. SSH is only needed if you need to connect to the host for troubleshooting.

IP address limitation

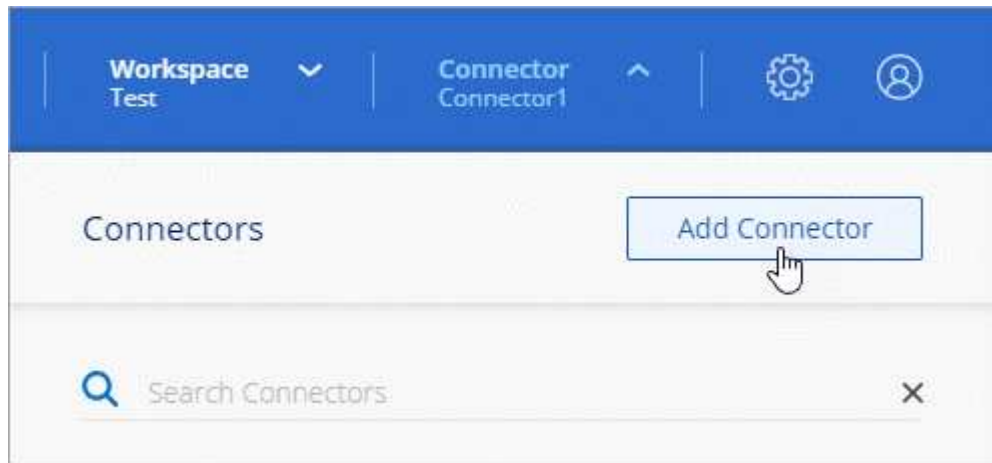
There's a possible conflict with IP addresses in the 172 range. [Learn more about this limitation.](#)

Create a Connector

BlueXP enables you to create a Connector in AWS directly from its user interface.

Steps

1. If you're creating your first Working Environment, click **Add Working Environment** and follow the prompts. Otherwise, click the **Connector** drop-down and select **Add Connector**.



2. Choose **Amazon Web Services** as your cloud provider and click **Continue**.
3. On the **Deploying a Connector** page, review the details about what you'll need. You have two options:
 - a. Click **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. Click **Skip to Deployment** if you already prepared by following the steps on this page.
4. Follow the steps in the wizard to create the Connector:
 - **Get Ready:** Review what you'll need.
 - **AWS Credentials:** Specify your AWS region and then choose an authentication method, which is either an IAM role that BlueXP 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 Connector 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.](#)

- **Details:** Provide details about the Connector.
 - Enter a name for the instance.
 - Add custom tags (metadata) to the instance.
 - Choose whether you want BlueXP 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 Connector's EBS disks. You have the option to use the default encryption key or to use a custom key.
- **Network:** Specify a VPC, subnet, and key pair for the instance, choose whether to enable a public IP address, and optionally specify a proxy configuration.

Make sure that you have the correct key pair to use with the Connector. Without a key pair, you will not be able to access the Connector virtual machine.

- **Security Group:** Choose whether to create a new security group or whether to select an existing security group that allows inbound HTTP, HTTPS, and SSH access.

- **Review:** Review your selections to verify that your set up is correct.

5. Click **Add**.

The instance should be ready in about 7 minutes. You should stay on the page until the process is complete.

After you finish

If you have Amazon S3 buckets in the same AWS account where you created the Connector, you'll see an Amazon S3 working environment appear on the Canvas automatically. [Learn more about what you can do with this working environment](#).

Open port 3128 for AutoSupport messages

If you plan to deploy Cloud Volumes ONTAP systems in a subnet where an outbound internet connection won't be available, then BlueXP automatically configures Cloud Volumes ONTAP to use the Connector as a proxy server.

The only requirement is to ensure that the Connector's security group allows *inbound* connections over port 3128. You'll need to open this port after you deploy the Connector.

If you use the default security group for Cloud Volumes ONTAP, then no changes are needed to its security group. But if you plan to define strict outbound rules for Cloud Volumes ONTAP, then you'll also need to ensure that the Cloud Volumes ONTAP security group allows *outbound* connections over port 3128.

Create a Connector in Azure from BlueXP

A BlueXP Account Admin needs to deploy a *Connector* before you can use most BlueXP features. The Connector enables BlueXP to manage resources and processes within your public cloud environment.

These steps describe how to create a Connector in an Azure commercial region directly from the BlueXP SaaS website.

- [Learn how to deploy a Connector in a Government region](#)
- [Learn about other ways to deploy a Connector](#)

Overview

To deploy a Connector, you need to provide BlueXP with a login that has the required permissions to create the Connector VM in Azure.

You have two options:

1. Sign in with your Microsoft account when prompted. This account must have specific Azure permissions. This is the default option.

[Follow the steps below to get started.](#)

2. Provide details about an Azure AD service principal. This service principal also requires specific permissions.

[Follow the steps below to get started.](#)

A note about Azure regions

The Connector 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](#)

Set up networking

Set up your networking so the Connector can manage resources and processes within your public cloud environment. Other than having a virtual network and subnet for the Connector, you'll need to ensure that the following requirements are met.


Connection to target networks

A Connector requires a network connection to the type of working environment that you're creating and the services that you're planning to enable.

For example, if you install a Connector in your corporate network, then you must set up a VPN connection to the virtual network in which you launch Cloud Volumes ONTAP.

Outbound internet access

The Connector requires outbound internet access to manage resources and processes within your public cloud environment.

Endpoints	Purpose
https://support.netapp.com	To obtain licensing information and to send AutoSupport messages to NetApp support.
https://*.api.blueexp.netapp.com https://api.blueexp.netapp.com https://*.cloudmanager.cloud.netapp.com https://cloudmanager.cloud.netapp.com	<div>The Connector is currently contacting "cloudmanager.cloud.netapp.com" but it will start contacting "api.blueexp.netapp.com" in an upcoming release.</div> To provide SaaS features and services within BlueXP.
https://cloudmanagerinfraprod.azurecr.io https://*.blob.core.windows.net	To upgrade the Connector and its Docker components.

Proxy server

If your organization requires deployment of an HTTP proxy for all outgoing internet traffic, obtain the following information about your HTTP proxy:

- IP address
- Credentials
- HTTPS certificate

Security group

There's no incoming traffic to the Connector, unless you initiate it or if the Connector is used as a proxy for AutoSupport messages. HTTP and HTTPS provide access to the [local UI](#), which you'll use in rare circumstances. SSH is only needed if you need to connect to the host for troubleshooting.

IP address limitation

There's a possible conflict with IP addresses in the 172 range. [Learn more about this limitation](#).

Create a Connector using your Azure account

The default way to create a Connector in Azure is by logging in with your Azure account when prompted. The login form is owned and hosted by Microsoft. Your credentials are not provided to NetApp.

Set up permissions for your Azure account

Before you can deploy a Connector from BlueXP, you need to ensure that your Azure account has the correct permissions.

Steps

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



This policy contains only the permissions needed to launch the Connector VM in Azure from BlueXP. Don't use this policy for other situations. When BlueXP creates the Connector, it applies a new set of permissions to the Connector VM that enables the Connector to manage the resources in your public cloud environment.

```
{
  "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/read",

"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

```

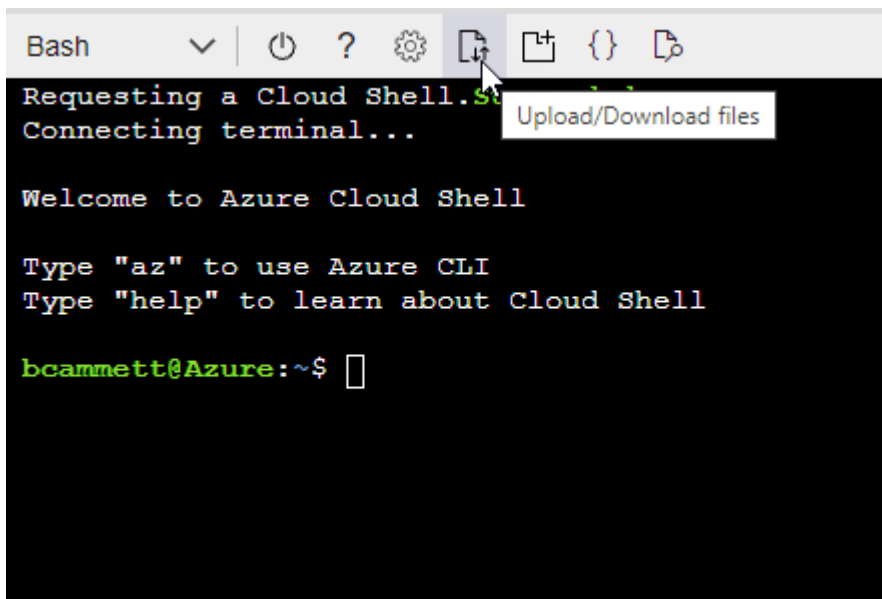
"AssignableScopes": [
  "/subscriptions/d333af45-0d07-4154-943d-c25fbzzzzzzz"
],

```

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 should now have a custom role called *Azure SetupAsService*.

4. Assign the role to the user who will deploy the Connector from BlueXP:
 - a. Open the **Subscriptions** service and select the user's subscription.
 - b. Click **Access control (IAM)**.
 - c. Click **Add > Add role assignment** and then add the permissions:
 - Select the **Azure SetupAsService** role and click **Next**.



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

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

Result

The Azure user now has the permissions required to deploy the Connector from BlueXP.

Create the Connector by logging in with your Azure account

BlueXP enables you to create a Connector in Azure directly from its user interface.

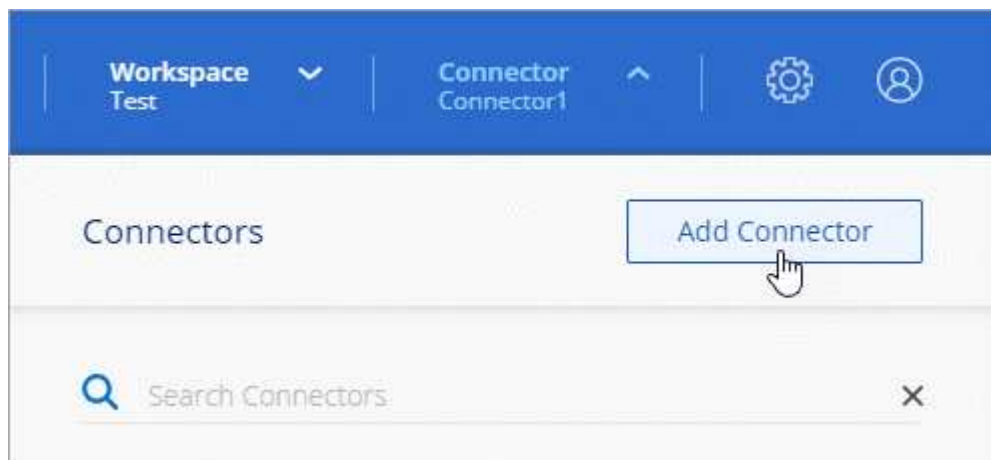
What you'll need

- An Azure subscription.
- A VNet and subnet in your Azure region of choice.
- If you don't want BlueXP to automatically create an Azure role for the Connector, then you'll need to create your own [using the policy on this page](#).

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

Steps

1. If you're creating your first Working Environment, click **Add Working Environment** and follow the prompts. Otherwise, click the **Connector** drop-down and select **Add Connector**.



2. Choose **Microsoft Azure** as your cloud provider.
3. On the **Deploying a Connector** page, review the details about what you'll need. You have two options:
 - a. Click **Continue** to prepare for deployment by using the in-product guide. Each step include information contained on this page of the documentation.
 - b. Click **Skip to Deployment** if you already prepared by following the steps on this page.
4. Follow the steps in the wizard to create the Connector:
 - If you're prompted, log in to your Microsoft account, which should have the required permissions to create the virtual machine.

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 BlueXP will automatically use that account. If you have multiple accounts, then you might need to log out first to ensure that you're using the right account.

- **VM Authentication:** Choose an Azure subscription, a location, a new resource group or an existing resource group, and then choose an authentication method.
- **Details:** Enter a name for the instance, specify tags, and choose whether you want BlueXP 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 subscriptions associated with this role. Each subscription that you choose provides the Connector with permissions to deploy Cloud Volumes ONTAP in those subscriptions.

- **Network:** Choose a VNet and subnet, whether to enable a public IP address, and optionally specify a proxy configuration.
- **Security Group:** Choose whether to create a new security group or whether to select an existing security group that allows inbound HTTP, HTTPS, and SSH access.
- **Review:** Review your selections to verify that your set up is correct.

5. Click **Add**.

The virtual machine should be ready in about 7 minutes. You should stay on the page until the process is complete.

After you finish

You need to associate a Connector with workspaces so Workspace Admins can use those Connectors to create Cloud Volumes ONTAP systems. If you only have Account Admins, then associating the Connector with workspaces isn't required. Account Admins have the ability to access all workspaces in BlueXP by default. [Learn more](#).

If you have Azure Blob storage in the same Azure account where you created the Connector, you'll see an Azure Blob working environment appear on the Canvas automatically. [Learn more about what you can do with this working environment](#).

Create a Connector using a service principal

Rather than logging in with your Azure account, you also have the option to provide BlueXP with the credentials for an Azure service principal that has the required permissions.

Granting Azure permissions using a service principal

Grant the required permissions to deploy a Connector in Azure by creating and setting up a service principal in Azure Active Directory and by obtaining the Azure credentials that BlueXP needs.

Steps

1. [Create an Azure Active Directory application](#).
2. [Assign the application to a role](#).
3. [Add Windows Azure Service Management API permissions](#).
4. [Get the application ID and directory ID](#).
5. [Create a client secret](#).

Create an Azure Active Directory application

Create an Azure Active Directory (AD) application and service principal that BlueXP can use to deploy the Connector.

Before you begin

You must have the right 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](#).

Steps

1. From the Azure portal, open the **Azure Active Directory** service.



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

Result

You've created the AD application and service principal.

Assign the application to a role

You must bind the service principal to the Azure subscription in which you plan to deploy the Connector and assign it the custom "Azure SetupAsService" role.

Steps

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



This policy contains only the permissions needed to launch the Connector VM in Azure from BlueXP. Don't use this policy for other situations. When BlueXP creates the Connector, it applies a new set of permissions to the Connector VM that enables the Connector to manage the resources in your public cloud environment.

```
{
  "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/read",

        "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 file by adding your Azure subscription ID to the assignable scope.

Example

```
"AssignableScopes": [  
  "/subscriptions/398e471c-3b42-4ae7-9b59-ce5bbzzzzzzz"
```

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 should now have a custom role called *Azure SetupAsService*.

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

Add role assignment ...

[Got feedback?](#)

Role **Members** Review + assign

Selected role Cloud Manager Operator 3.9.12_B

Assign access to ☒ User, group, or service principal ☐ Managed identity

Members [+ Select members](#)

- Search for the name of the application.

Here's an example:

Select members ✕

Select ⓘ

test-service-principal

test-service-principal

- Select the application and click **Select**.
 - Click **Next**.
- f. Click **Review + assign**.

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

Add Windows Azure Service Management API permissions

The service principal must have "Windows Azure Service Management API" permissions.

Steps

1. In the **Azure Active Directory** service, click **App registrations** and select the application.
2. Click **API permissions > Add a permission**.
3. Under **Microsoft APIs**, select **Azure Service Management**.

Request API permissions

Select an API

[Microsoft APIs](#) [APIs my organization uses](#) [My APIs](#)

Commonly used Microsoft APIs

Microsoft Graph
Take advantage of the tremendous amount of data in Office 365, Enterprise Mobility + Security, and Windows 10. Access Azure AD, Excel, Intune, Outlook/Exchange, OneDrive, OneNote, SharePoint, Planner, and more through a single endpoint.



**Azure Batch**
Schedule large-scale parallel and HPC applications in the cloud

**Azure Data Catalog**
Programmatic access to Data Catalog resources to register, annotate and search data assets

**Azure Data Explorer**
Perform ad-hoc queries on terabytes of data to build near real-time and complex analytics solutions

**Azure Data Lake**
Access to storage and compute for big data analytic scenarios

**Azure DevOps**
Integrate with Azure DevOps and Azure DevOps server

**Azure Import/Export**
Programmatic control of import/export jobs

**Azure Key Vault**
Manage your key vaults as well as the keys, secrets, and certificates within your Key Vaults

**Azure Rights Management Services**
Allow validated users to read and write protected content

**Azure Service Management**
Programmatic access to much of the functionality available through the Azure portal

**Azure Storage**
Secure, massively scalable object and data lake storage for unstructured and semi-structured data

**Customer Insights**
Create profile and interaction models for your products

**Data Export Service for Microsoft Dynamics 365**
Export data from Microsoft Dynamics CRM organization to an external destination

4. Click **Access Azure Service Management as organization users** and then click **Add permissions**.

Request API permissions

[← All APIs](#)



Azure Service Management

<https://management.azure.com/> [Docs](#)

What type of permissions does your application require?

Delegated permissions

Your application needs to access the API as the signed-in user.

Application permissions

Your application runs as a background service or daemon without a signed-in user.

Select permissions

[expand all](#)

Type to search	
PERMISSION	ADMIN CONSENT REQUIRED
<input checked="" type="checkbox"/> user_impersonation Access Azure Service Management as organization users (preview) ⓘ	-

Get the application ID and directory ID

When you create the Connector from BlueXP, you need to provide the application (client) ID and the directory (tenant) ID for the application. BlueXP uses the IDs to programmatically sign in.

Steps

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



Create a client secret

You need to create a client secret and then provide BlueXP with the value of the secret so BlueXP can use it to authenticate with Azure AD.

Steps

1. Open the **Azure Active Directory** service.
2. Click **App registrations** and select your application.
3. Click **Certificates & secrets > New client secret**.

4. Provide a description of the secret and a duration.
5. Click **Add**.
6. Copy the value of the client secret.

Client secrets

A secret string that the application uses to prove its identity when requesting a token. Also can be referred to as application password.

[+ New client secret](#)

DESCRIPTION	EXPIRES	VALUE	Copy to clipboard
test secret	8/16/2020	*sZ1jSe2By:D*-ZRoV4NLfdAcY7:+0vA	

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 BlueXP when you create the Connector.

Create the Connector by logging in with the service principal

BlueXP enables you to create a Connector in Azure directly from its user interface.

What you'll need

- An Azure subscription.
- A VNet and subnet in your Azure region of choice.
- Details about an HTTP proxy, if your organization requires a proxy for all outgoing internet traffic:
 - IP address
 - Credentials
 - HTTPS certificate
- If you don't want BlueXP to automatically create an Azure role for the Connector, then you'll need to create your own [using the policy on this page](#).

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

Steps

1. If you're creating your first Working Environment, click **Add Working Environment** and follow the prompts. Otherwise, click the **Connector** drop-down and select **Add Connector**.



2. Choose **Microsoft Azure** as your cloud provider.
3. On the **Deploying a Connector** page:
 - a. Under **Authentication**, click **Active Directory service principal** and enter information about the Azure Active Directory service principal that grants the required permissions:
 - Application (client) ID: See [Get the application ID and directory ID](#).
 - Directory (tenant) ID: See [Get the application ID and directory ID](#).
 - Client Secret: See [Create a client secret](#).
 - b. Click **Log in**.
 - c. You now have two options:
 - Click **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.
 - Click **Skip to Deployment** if you already prepared by following the steps on this page.
4. Follow the steps in the wizard to create the Connector:
 - **VM Authentication:** Choose an Azure subscription, a location, a new resource group or an existing resource group, and then choose an authentication method.
 - **Details:** Enter a name for the instance, specify tags, and choose whether you want BlueXP 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 subscriptions associated with this role. Each subscription that you choose provides the Connector with permissions to deploy Cloud Volumes ONTAP in those subscriptions.

 - **Network:** Choose a VNet and subnet, whether to enable a public IP address, and optionally specify a proxy configuration.
 - **Security Group:** Choose whether to create a new security group or whether to select an existing security group that allows inbound HTTP, HTTPS, and SSH access.
 - **Review:** Review your selections to verify that your set up is correct.
5. Click **Add**.

The virtual machine should be ready in about 7 minutes. You should stay on the page until the process is complete.

After you finish

You need to associate a Connector with workspaces so Workspace Admins can use those Connectors to create Cloud Volumes ONTAP systems. If you only have Account Admins, then associating the Connector with workspaces isn't required. Account Admins have the ability to access all workspaces in BlueXP by default. [Learn more.](#)

If you have Azure Blob storage in the same Azure account where you created the Connector, you'll see an Azure Blob working environment appear on the Canvas automatically. [Learn more about what you can do with this working environment.](#)

Open port 3128 for AutoSupport messages

If you plan to deploy Cloud Volumes ONTAP systems in a subnet where an outbound internet connection won't be available, then BlueXP automatically configures Cloud Volumes ONTAP to use the Connector as a proxy server.

The only requirement is to ensure that the Connector's security group allows *inbound* connections over port 3128. You'll need to open this port after you deploy the Connector.

If you use the default security group for Cloud Volumes ONTAP, then no changes are needed to its security group. But if you plan to define strict outbound rules for Cloud Volumes ONTAP, then you'll also need to ensure that the Cloud Volumes ONTAP security group allows *outbound* connections over port 3128.

Create a Connector in Google Cloud from BlueXP

A BlueXP Account Admin needs to deploy a *Connector* before you can use most BlueXP features. [Learn when a Connector is required.](#) The Connector enables BlueXP to manage resources and processes within your public cloud environment.

This page describes how to create a Connector in Google Cloud directly from BlueXP. [Learn about other ways to deploy a Connector.](#)

These steps must be completed by a user who has the Account Admin role. A Workspace Admin can't create a Connector.



When you create your first Cloud Volumes ONTAP working environment, BlueXP will prompt you to create a Connector if you don't have one yet.

Quick start

Get started quickly by following these steps, or scroll down to the remaining sections for full details.

1

Set up permissions

- Ensure that your Google Cloud account has the correct permissions by creating and attaching a custom role.

[Set up permissions to deploy the Connector.](#)

- When you create the Connector VM, you need to associate it with a service account. This service account must have a custom role that has permissions to manage resources in Google Cloud.

[Set up a service account for the Connector.](#)

- If you're deploying Cloud Volumes ONTAP across projects, ensure that the Connector has access to those projects.

[Set up permissions across projects.](#)

- If you're using a shared VPC, set up permissions in the service project and host project.

[Set up shared VPC permissions.](#)

2

Set up networking

You'll need a VPC and subnet with outbound internet access to specific endpoints. If an HTTP proxy is required for outbound internet, then you'll need the IP address, credentials, and HTTPS certificate.

[View networking requirements.](#)

3

Enable Google Cloud APIs

- Cloud Deployment Manager V2 API
- Cloud Logging API
- Cloud Resource Manager API
- Compute Engine API
- Identity and Access Management (IAM) API

4

Create the Connector

Click the Connector drop-down, select **Add Connector** and follow the prompts.

[Follow step-by-step instructions.](#)

Set up permissions

Permissions are required for the following:

- The user who will deploy the Connector VM
- A service account that you need to attach to the Connector VM during deployment

Depending on your configuration, you might need to complete the following steps as well:

- Set up permissions across projects
- Set up permissions for a shared VPC

Set up permissions to deploy the Connector

Before you can deploy a Connector, you need to ensure that your Google Cloud account has the correct permissions.

Steps

1. [Create a custom role](#) that includes the following permissions:

```
title: Connector deployment policy
description: Permissions for the user who deploys the Connector from
BlueXP
stage: GA
includedPermissions:
- 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
- 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.list
```

2. Attach the custom role to the user who will deploy the Connector from BlueXP.

Result

The Google Cloud user now has the permissions required to create the Connector.

Set up a service account for the Connector

A service account is required to provide the Connector with the permission that it needs to manage resources in Google Cloud. You'll associate this service account with the Connector VM when you create it.

The permissions for the service account are different than the permissions that you set up in the previous section.

Steps

1. [Create a custom role](#) that includes the following permissions:

```
title: NetApp BlueXP
description: Permissions for the service account associated with the
Connector instance.
stage: GA
includedPermissions:
- iam.serviceAccounts.actAs
- compute.regionBackendServices.create
- compute.regionBackendServices.get
- compute.regionBackendServices.list
- compute.networks.updatePolicy
- compute.backendServices.create
- compute.addresses.list
```


- `compute.disks.create`
- `compute.disks.createSnapshot`
- `compute.disks.delete`
- `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.addAccessConfig`
- `compute.instances.attachDisk`
- `compute.instances.create`
- `compute.instances.delete`
- `compute.instances.detachDisk`
- `compute.instances.get`
- `compute.instances.getSerialPortOutput`
- `compute.instances.list`
- `compute.instances.setDeletionProtection`
- `compute.instances.setLabels`
- `compute.instances.setMachineType`
- `compute.instances.setMetadata`
- `compute.instances.setTags`
- `compute.instances.start`
- `compute.instances.stop`
- `compute.instances.updateDisplayDevice`
- `compute.machineTypes.get`
- `compute.networks.get`
- `compute.networks.list`
- `compute.projects.get`
- `compute.regions.get`
- `compute.regions.list`
- `compute.snapshots.create`
- `compute.snapshots.delete`
- `compute.snapshots.get`
- `compute.snapshots.list`
- `compute.snapshots.setLabels`
- `compute.subnetworks.get`
- `compute.subnetworks.list`
- `compute.subnetworks.use`

```
- compute.subnetworks.useExternalIp
- compute.zoneOperations.get
- compute.zones.get
- compute.zones.list
- compute.instances.setServiceAccount
- 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
- logging.logEntries.list
- logging.privateLogEntries.list
- resourcemanager.projects.get
- storage.buckets.create
- storage.buckets.delete
- storage.buckets.get
- storage.buckets.list
- cloudkms.cryptoKeyVersions.useToEncrypt
- cloudkms.cryptoKeys.get
- cloudkms.cryptoKeys.list
- cloudkms.keyRings.list
- storage.buckets.update
- iam.serviceAccounts.getIamPolicy
- iam.serviceAccounts.list
- storage.objects.get
- storage.objects.list
- monitoring.timeSeries.list
- storage.buckets.getIamPolicy
```

2. Create a Google Cloud service account and apply the custom role that you just created.
3. If you want to deploy Cloud Volumes ONTAP in other projects, [grant access by adding the service account with the BlueXP role to that project](#). You'll need to repeat this step for each project.

Result

The service account for the Connector VM is set up.

Set up permissions across projects

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

For example, let's say the Connector 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.

Steps

1. In the Google Cloud console, go to the IAM service and select the project where you want to create Cloud Volumes ONTAP systems.
2. On the **IAM** page, select **Grant Access** and provide the required details.
 - Enter the email of the Connector's service account.
 - Select the Connector's custom role.
 - Click **Save**.

For more details, refer to [Google Cloud documentation](#)

Set up shared VPC permissions

If you are using a shared VPC to deploy resources into a service project, then the following permissions are required. This table is for reference and your environment should reflect the permissions table when IAM configuration is complete.

Identity	Creator	Hosted in	Service project permissions	Host project permissions	Purpose
Google account used to deploy the Connector	Custom	Service Project	<ul style="list-style-type: none">• The permissions found in this section above	<ul style="list-style-type: none">• compute.networkUser	Deploying the Connector in the service project
Connector service account	Custom	Service project	<ul style="list-style-type: none">• The permissions found in this section above	<ul style="list-style-type: none">• compute.networkUser• deploymentmanager.editor	Deploying and maintaining Cloud Volumes ONTAP and services in the service project
Cloud Volumes ONTAP service account	Custom	Service project	<ul style="list-style-type: none">• storage.admin• member: BlueXP service account as serviceAccount.user	N/A	(Optional) For data tiering and Cloud Backup

Identity	Creator	Hosted in	Service project permissions	Host project permissions	Purpose
Google APIs service agent	Google Cloud	Service project	• (Default) Editor	• compute.networkUser	Interacts with Google Cloud APIs on behalf of deployment. Allows BlueXP 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 BlueXP 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 BlueXP create them for you. BlueXP will create 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 BlueXP create them for you. These permissions reside in the BlueXP 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 data 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.

Set up networking

Set up your networking so the Connector can manage resources and processes within your public cloud environment. Other than having a virtual network and subnet for the Connector, you'll need to ensure that the following requirements are met.

Connection to target networks


A Connector requires a network connection to the type of working environment that you're creating and the services that you're planning to enable.

For example, if you install a Connector in your corporate network, then you must set up a VPN connection to the virtual network in which you launch Cloud Volumes ONTAP.

Outbound internet access

The Connector requires outbound internet access to manage resources and processes within your public cloud environment.

Endpoints	Purpose
https://support.netapp.com	To obtain licensing information and to send AutoSupport messages to NetApp support.

Endpoints	Purpose
https://*.api.bluexp.netapp.com https://api.bluexp.netapp.com https://*.cloudmanager.cloud.netapp.com https://cloudmanager.cloud.netapp.com	To provide SaaS features and services within BlueXP. <div>  <p>The Connector is currently contacting "cloudmanager.cloud.netapp.com" but it will start contacting "api.bluexp.netapp.com" in an upcoming release.</p> </div>
https://cloudmanagerinfraprod.azurecr.io https://*.blob.core.windows.net	To upgrade the Connector and its Docker components.

Proxy server

If your organization requires deployment of an HTTP proxy for all outgoing internet traffic, obtain the following information about your HTTP proxy:

- IP address
- Credentials
- HTTPS certificate

Security group

There's no incoming traffic to the Connector, unless you initiate it or if the Connector is used as a proxy for AutoSupport messages. HTTP and HTTPS provide access to the [local UI](#), which you'll use in rare circumstances. SSH is only needed if you need to connect to the host for troubleshooting.

IP address limitation

There's a possible conflict with IP addresses in the 172 range. [Learn more about this limitation.](#)

Enable Google Cloud APIs

Several APIs are required to deploy the Connector and Cloud Volumes ONTAP.

Step

1. [Enable the following Google Cloud APIs in your project.](#)
 - Cloud Deployment Manager V2 API
 - Cloud Logging API
 - Cloud Resource Manager API
 - Compute Engine API
 - Identity and Access Management (IAM) API

Create a Connector

Create a Connector in Google Cloud directly from the BlueXP user interface or by using gcloud.

BlueXP

1. If you're creating your first Working Environment, click **Add Working Environment** and follow the prompts. Otherwise, click the **Connector** drop-down and select **Add Connector**.



2. Choose **Google Cloud Platform** as your cloud provider.
3. On the **Deploying a Connector** page, review the details about what you'll need. You have two options:
 - a. Click **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. Click **Skip to Deployment** if you already prepared by following the steps on this page.
4. Follow the steps in the wizard to create the Connector:
 - 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.
 - **Firewall Policy:** Choose whether to create a new firewall policy or whether to select an existing firewall policy that allows inbound HTTP, HTTPS, and SSH access.
 - **Review:** Review your selections to verify that your set up is correct.
5. Click **Add**.

The instance should be ready in about 7 minutes. You should stay on the page until the process is complete.

gcloud

1. Log in to the gcloud SDK using your preferred methodology.

In our examples, we'll use a local shell with the gcloud SDK installed, but you could use the native

Google Cloud Shell in the Google Cloud console.

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-4
  --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 Connector instance

network-path

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

subnet-path

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

kms-key-path

(Optional) Add a KMS key to encrypt the Connector'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 Connector using the NetApp golden image. The Connector instance and software should be running in approximately five minutes.

4. Open a web browser from a host that has a connection to the Connector instance and enter the following URL:

`https://ipaddress`

5. After you log in, set up the Connector:
 - a. Specify the NetApp account to associate with the Connector.

[Learn about NetApp accounts](#).

- b. Enter a name for the system.

Result

The Connector is now installed and set up with your NetApp account. BlueXP will automatically use this Connector when you create new working environments. But if you have more than one Connector, you'll need to [switch between them](#).

If you have Google Cloud Storage buckets in the same Google Cloud account where you created the Connector, you'll see a Google Cloud Storage working environment appear on the Canvas automatically. [Learn more about what you can do with this working environment](#).

Open port 3128 for AutoSupport messages

If you plan to deploy Cloud Volumes ONTAP systems in a subnet where an outbound internet connection won't be available, then BlueXP automatically configures Cloud Volumes ONTAP to use the Connector as a proxy server.

The only requirement is to ensure that the Connector's security group allows *inbound* connections over port 3128. You'll need to open this port after you deploy the Connector.

If you use the default security group for Cloud Volumes ONTAP, then no changes are needed to its security group. But if you plan to define strict outbound rules for Cloud Volumes ONTAP, then you'll also need to ensure that the Cloud Volumes ONTAP security group allows *outbound* connections over port 3128.

Create a Connector in a Government region

If you are operating in a Government region, you need to deploy a Connector from your cloud provider's marketplace or by manually installing the Connector software on an existing Linux host. You can't deploy the Connector in a Government region from BlueXP's SaaS website.

Use one of the following links to view instructions for creating a Connector:

- [Create a Connector from the AWS Marketplace](#)
- [Create a Connector and Cloud Volumes ONTAP in the AWS C2S environment](#)
- [Create a Connector from the Azure Marketplace](#)
- [Install a Connector on your own Linux host](#)

For manual installations on your own Linux host, you must use the "online" installer to install the Connector on a host that has internet access. A separate "offline" installer is available for the Connector, but it's only supported with on-prem sites that don't have internet access. It's not supported with Government regions.

After you deploy the Connector, you can access BlueXP by opening your web browser and connecting to the IP address of the Connector instance: `https://ipaddress`

Since the Connector was deployed in a Government region, it's not accessible from <https://console.bluexp.netapp.com>.

Copyright information

Copyright © 2023 NetApp, Inc. All Rights Reserved. Printed in the U.S. No part of this document covered by copyright may be reproduced in any form or by any means—graphic, electronic, or mechanical, including photocopying, recording, taping, or storage in an electronic retrieval system—without prior written permission of the copyright owner.

Software derived from copyrighted NetApp material is subject to the following license and disclaimer:

THIS SOFTWARE IS PROVIDED BY NETAPP “AS IS” AND WITHOUT ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE, WHICH ARE HEREBY DISCLAIMED. IN NO EVENT SHALL NETAPP BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

NetApp reserves the right to change any products described herein at any time, and without notice. NetApp assumes no responsibility or liability arising from the use of products described herein, except as expressly agreed to in writing by NetApp. The use or purchase of this product does not convey a license under any patent rights, trademark rights, or any other intellectual property rights of NetApp.

The product described in this manual may be protected by one or more U.S. patents, foreign patents, or pending applications.

LIMITED RIGHTS LEGEND: Use, duplication, or disclosure by the government is subject to restrictions as set forth in subparagraph (b)(3) of the Rights in Technical Data -Noncommercial Items at DFARS 252.227-7013 (FEB 2014) and FAR 52.227-19 (DEC 2007).

Data contained herein pertains to a commercial product and/or commercial service (as defined in FAR 2.101) and is proprietary to NetApp, Inc. All NetApp technical data and computer software provided under this Agreement is commercial in nature and developed solely at private expense. The U.S. Government has a non-exclusive, non-transferrable, nonsublicensable, worldwide, limited irrevocable license to use the Data only in connection with and in support of the U.S. Government contract under which the Data was delivered. Except as provided herein, the Data may not be used, disclosed, reproduced, modified, performed, or displayed without the prior written approval of NetApp, Inc. United States Government license rights for the Department of Defense are limited to those rights identified in DFARS clause 252.227-7015(b) (FEB 2014).

Trademark information

NETAPP, the NETAPP logo, and the marks listed at <http://www.netapp.com/TM> are trademarks of NetApp, Inc. Other company and product names may be trademarks of their respective owners.