Set up your cloud provider

Astra Control Service

NetApp
November 09, 2021

Table of Contents

Set up your cloud provider

Set up Google Cloud

A few steps are required to prepare your Google Cloud project before you can manage Google Kubernetes Engine clusters with Astra Control Service.

Quick start for setting up Google Cloud

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

[One] Review Astra Control Service requirements for Google Kubernetes Engine

Ensure that clusters are healthy and running a Kubernetes version in the range of 1.18 to 1.20, that worker nodes are online and running Container-Optimized OS or Ubuntu, and more. Learn more about this step.

[Two] Purchase Cloud Volumes Service for Google Cloud

Go to the NetApp Cloud Volumes Service page in the Google Cloud Marketplace and click Purchase. Learn more about this step.

[Three] Enable APIs in your Google Cloud project

Enable the following Google Cloud APIs:

- Google Kubernetes Engine
- Cloud Storage
- Cloud Storage JSON API
- Service Usage
- Cloud Resource Manager API
- NetApp Cloud Volumes Service
- Service Consumer Management API
- Service Networking API
- Service Management API

Follow step-by-step instructions.

[Four] Create a service account that has the required permissions

Create a Google Cloud service account that has the following permissions:

- Kubernetes Engine Admin
- NetApp Cloud Volumes Admin
- Storage Admin
- Service Usage Viewer
- Compute Network Viewer

Read step-by-step instructions.

[Five] Create a service account key

Create a key for the service account and save the key file in a secure location. Follow step-by-step instructions.

[Six] Set up network peering for your VPC

Set up network peering from your VPC to Cloud Volumes Service for Google Cloud. Follow step-by-step instructions.

The following image depicts each of these steps that you'll need to complete.

GKE cluster requirements

A Kubernetes cluster must meet the following requirements so you can discover and manage it from Astra Control Service.

Kubernetes version
: A cluster must be running a Kubernetes version in the range of 1.18 to 1.20.

Image type
: The image type for each worker node must be one of the following:

  - Container-Optimized OS with Containerd
  - Container-Optimized OS with Docker
  - Ubuntu with Docker

Cluster state
: Clusters must be running in a healthy state and have at least one online worker node with no worker nodes in a failed state.

Google Cloud region
: Clusters must be running in a Google Cloud region where Cloud Volumes Service for Google Cloud is supported. Note that Astra Control Service supports both service types: CVS and CVS-Performance.

Networking
: The cluster must reside in a VPC that is peered with Cloud Volumes Service for Google Cloud. This step is described below.

Private clusters
: If the cluster is private, the authorized networks must allow the Astra Control Service IP addresses:

  - 54.164.233.140/32
  - 3.218.120.204/32
  - 34.193.99.138/32

Mode of operation for a GKE cluster
: You should use the Standard mode of operation. The Autopilot mode hasn't been tested at this time. Learn more about modes of operation.

Purchase Cloud Volumes Service for Google Cloud

Astra Control Service uses Cloud Volumes Service for Google Cloud as the backend storage for your persistent volumes. You need to purchase Cloud Volumes Service for Google Cloud from the Google Cloud Marketplace to enable billing for persistent volumes.

Step

1. Go to the NetApp Cloud Volumes Service page in the Google Cloud Marketplace, click **Purchase**, and follow the prompts.

   Follow step-by-step instructions in the Google Cloud documentation to purchase and enable the service.

Enable APIs in your project

Your project needs permissions to access specific Google Cloud APIs. APIs are used to interact with Google Cloud resources, such as Google Kubernetes Engine (GKE) clusters and NetApp Cloud Volumes Service storage.

Step

1. Use the Google Cloud console or gcloud CLI to enable the following APIs:
   - Google Kubernetes Engine
   - Cloud Storage
   - Cloud Storage JSON API
   - Service Usage
   - Cloud Resource Manager API
   - NetApp Cloud Volumes Service
   - Service Consumer Management API
   - Service Networking API
   - Service Management API

The following video shows how to enable the APIs from the Google Cloud console.

► https://docs.netapp.com/us-en/astra-control-service/media/get-started/video-enable-gcp-apis.mp4 *(video)*

Create a service account

Astra Control Service uses a Google Cloud service account to facilitate Kubernetes application data management on your behalf.

Steps

1. Go to Google Cloud and create a service account by using the console, gcloud command, or another preferred method.
2. Grant the service account the following roles:
   - **Kubernetes Engine Admin** - Used to list clusters and create admin access to manage apps.
   - **NetApp Cloud Volumes Admin** - Used to manage persistent storage for apps.
   - **Storage Admin** - Used to manage buckets and objects for backups of apps.

- **Service Usage Viewer** - Used to check if the required Cloud Volumes Service for Google Cloud APIs are enabled.
- **Compute Network Viewer** - Used to check if the Kubernetes VPC is allowed to reach Cloud Volumes Service for Google Cloud.

If you'd like to use gcloud, you can follow steps from within the Astra Control interface. Click **Account > Credentials > Add Credentials**, and then click **Instructions**.

If you'd like to use the Google Cloud console, the following video shows how to create the service account from the console.

► https://docs.netapp.com/us-en/astra-control-service/media/get-started/video-create-gcp-service-

account.mp4 *(video)*

Configure the service account for a shared VPC

To manage GKE clusters that reside in one project, but use a VPC from a different project (a shared VPC), then you need to specify the Astra service account as a member of the host project with the **Compute Network Viewer** role.

Steps

1. From the Google Cloud console, go to **IAM & Admin** and select **Service Accounts**.
2. Find the Astra service account that has the required permissions and then copy the email address.
3. Go to your host project and then select **IAM & Admin** > **IAM**.
4. Click **Add** and add an entry for the service account.
    a. **New members**: Enter the email address for the service account.
    b. **Role**: Select **Compute Network Viewer**.
    c. Click **Save**.

Result

Adding a GKE cluster using a shared VPC will fully work with Astra.

Create a service account key

Instead of providing a user name and password to Astra Control Service, you'll provide a service account key when you add your first cluster. Astra Control Service uses the service account key to establish the identity of the service account that you just set up.

The service account key is plaintext stored in the JavaScript Object Notation (JSON) format. It contains information about the GCP resources that you have permission to access.

You can only view or download the JSON file when you create the key. However, you can create a new key at any time.

Steps

1. Go to Google Cloud and create a service account key by using the console, gcloud command, or another preferred method.
2. When prompted, save the service account key file in a secure location.

The following video shows how to create the service account key from the Google Cloud console.

► https://docs.netapp.com/us-en/astra-control-service/media/get-started/video-create-gcp-service-account-

key.mp4 *(video)*

Set up network peering for your VPC

The final step is to set up networking peering from your VPC to Cloud Volumes Service for Google Cloud.

The easiest way to set up network peering is by obtaining the gcloud commands directly from Cloud Volumes Service. The commands are available from Cloud Volumes Service when creating a new file system.

Steps

1. Go to NetApp Cloud Central's Global Regions Maps and identify the service type that you'll be using in the Google Cloud region where your cluster resides.

   Cloud Volumes Service provides two service types: CVS and CVS-Performance. Learn more about these service types.

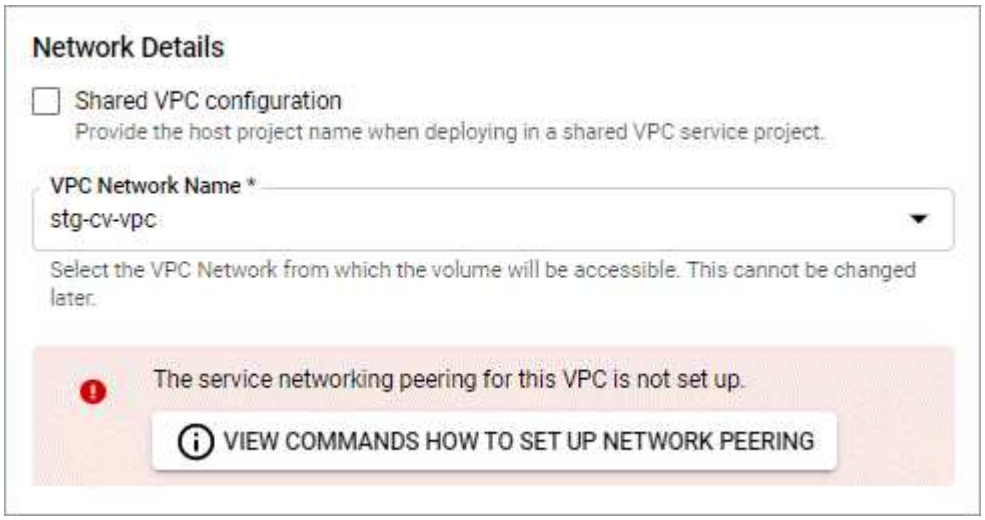
2. Go to Cloud Volumes in Google Cloud Platform.
3. On the **Volumes** page, click **Create**.
4. Under **Service Type**, select either **CVS** or **CVS-Performance**.

   You need to choose the correct service type for your Google Cloud region. This is the service type that you identified in step 1. After you select a service type, the list of regions on the page updates with the regions where that service type is supported.

   After this step, you'll only need to enter your networking information to obtain the commands.

5. Under **Region**, select your region and zone.
6. Under **Network Details**, select your VPC.

   If you haven't set up network peering, you'll see the following notification:



7. Click the button to view the network peering set up commands.
8. Copy the commands and run them in Cloud Shell.

   For more details about using these commands, refer to the Quickstart for Cloud Volumes Service for GCP.

   Learn more about configuring private services access and setting up network peering.

9. After you're done, you can click cancel on the **Create File System** page.

   We started creating this volume only to get the commands for network peering.

Set up Microsoft Azure

A few steps are required to prepare your Microsoft Azure subscription before you can manage Azure Kubernetes Service clusters with Astra Control Service.

Quick start for setting up Azure

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

[One] Review Astra Control Service requirements for Azure Kubernetes Service

Ensure that clusters are healthy and running Kubernetes version 1.17 or later, that node pools are online and running **Linux**, and more. Learn more about this step.

[Two] Register for Azure NetApp Files

Register the NetApp Resource Provider. Learn more about this step.

[Three] Create a NetApp account

In the Azure portal, go to Azure NetApp Files and create a NetApp account. Learn more about this step.

[Four] Set up capacity pools

Set up one or more capacity pools for your persistent volumes. Learn more about this step.

[Five] Delegate a subnet to Azure NetApp Files

Delegate a subnet to Azure NetApp Files so that Astra Control Service can create persistent volumes in that subnet. Learn more about this step.

[Six] Create an Azure service principal

Create an Azure service principal that has the Contributor role. Read step-by-step instructions.

AKS cluster requirements

A Kubernetes cluster must meet the following requirements so you can discover and manage it from Astra Control Service.

Kubernetes version

Clusters must be running Kubernetes version 1.17 or later.

Image type

The image type for all node pools must be Linux.

Cluster state

Clusters must be running in a healthy state and have at least one online worker node with no worker nodes in a failed state.

Azure region

Clusters must reside in a region where Azure NetApp Files is available. View Azure products by region.

Subscription

Clusters must reside in a subscription where Azure NetApp Files is enabled. You'll choose a subscription when you register for Azure NetApp Files.

VNet

- Clusters must reside in a VNet that has direct access to an Azure NetApp Files delegated subnet. Learn how to set up a delegated subnet.
- If your Kubernetes clusters are in a VNet that's peered to the Azure NetApp Files delegated subnet that's in another VNet, then both sides of the peering connection must be online.
- Be aware that the default limit for the number of IPs used in a VNet (including immediately peered VNets) with Azure NetApp Files is 1,000. View Azure NetApp Files resource limits.

  If you're close to the limit, you have two options:

  - You can submit a request for a limit increase. Contact your NetApp representative if you need help.
  - When creating a new AKS cluster, specify a new network for the cluster. Once the new network is created, provision a new subnet and delegate the subnet to Azure NetApp Files.

Private networking

Private networking must not be enabled on a cluster.

External volume snapshot controller

Clusters must have a CSI volume snapshot controller installed. This controller is installed by default starting with K8s version 1.21, but you'll need to check on clusters running versions 1.17, 1.18, 1.19, or 1.20. Learn more about an external snapshot controller for on-demand volume snapshots.

Install a CSI volume snapshot controller

As noted in the list of requirements, Kubernetes clusters must have a CSI volume snapshot controller installed. Follow these steps to install the controller on your clusters.

Steps for K8s versions 1.17, 1.18, and 1.19

1. Install volume snapshot CRDs.

```
kubectl apply -f https://raw.githubusercontent.com/kubernetes-
csi/external-snapshotter/release-
3.0/client/config/crd/snapshot.storage.k8s.io_volumesnapshotclasses.yaml
kubectl apply -f https://raw.githubusercontent.com/kubernetes-
csi/external-snapshotter/release-
3.0/client/config/crd/snapshot.storage.k8s.io_volumesnapshotcontents.yam
l
kubectl apply -f https://raw.githubusercontent.com/kubernetes-
csi/external-snapshotter/release-
3.0/client/config/crd/snapshot.storage.k8s.io_volumesnapshots.yaml
```

2. Create the snapshot controller.

   If you want the snapshot controller in a specific namespace, download and edit the following files before you apply them.

```
kubectl apply -f https://raw.githubusercontent.com/kubernetes-csi/external-snapshotter/release-3.0/deploy/kubernetes/snapshot-controller/rbac-snapshot-controller.yaml
kubectl apply -f https://raw.githubusercontent.com/kubernetes-csi/external-snapshotter/release-3.0/deploy/kubernetes/snapshot-controller/setup-snapshot-controller.yaml
```

Steps for K8s version 1.20

1. Install volume snapshot CRDs.

```
kubectl apply -f https://raw.githubusercontent.com/kubernetes-csi/external-snapshotter/v4.0.0/client/config/crd/snapshot.storage.k8s.io_volumesnapshotclasses.yaml
kubectl apply -f https://raw.githubusercontent.com/kubernetes-csi/external-snapshotter/v4.0.0/client/config/crd/snapshot.storage.k8s.io_volumesnapshotcontents.yaml
kubectl apply -f https://raw.githubusercontent.com/kubernetes-csi/external-snapshotter/v4.0.0/client/config/crd/snapshot.storage.k8s.io_volumesnapshots.yaml
```

2. Create the snapshot controller.

   If you want the snapshot controller in a specific namespace, download and edit the following files before you apply them.

```
kubectl apply -f https://raw.githubusercontent.com/kubernetes-csi/external-snapshotter/v4.0.0/deploy/kubernetes/snapshot-controller/rbac-snapshot-controller.yaml
kubectl apply -f https://raw.githubusercontent.com/kubernetes-csi/external-snapshotter/v4.0.0/deploy/kubernetes/snapshot-controller/setup-snapshot-controller.yaml
```

Register for Azure NetApp Files

If you don’t have a Microsoft Azure account, begin by signing up for Microsoft Azure. Then, get access to Azure NetApp Files by registering the NetApp Resource Provider.

Steps

1. If you already have a Microsoft Azure account, skip to step 4.
2. Go to the Azure subscription page to subscribe to the Azure service.
3. Select a plan and follow the instructions to complete the subscription.
4. Log in to the Azure portal.
5. Follow Azure NetApp Files documentation to register the NetApp Resource Provider.

Create a NetApp account

After you've registered, create a NetApp account in Azure NetApp Files.

Step

1. Follow Azure NetApp Files documentation to create a NetApp account from the Azure portal.

Set up a capacity pool

One or more capacity pools are required so that Astra Control Service can provision persistent volumes in a capacity pool. Astra Control Service doesn't create capacity pools for you.

Take the following into consideration as you set up capacity pools for your Kubernetes apps:

- A capacity pool can have an Ultra, Premium, or Standard service level. Each of these service levels are designed for different performance needs. Astra Control Service supports all three.

  You need to set up a capacity pool for each service level that you want to use with your Kubernetes clusters.

  Learn more about service levels for Azure NetApp Files.

- Before you create a capacity pool for the apps that you intend to protect with Astra Control Service, choose the required performance and capacity for those apps.

  Provisioning the right amount of capacity ensures that users can create persistent volumes as they are needed. If capacity isn't available, then the persistent volumes can't be provisioned.

- An Azure NetApp Files capacity pool can use the manual or auto QoS type. Astra Control Service supports auto QoS capacity pools. Manual QoS capacity pools aren't supported.

Step

1. Follow Azure NetApp Files documentation to set up an auto QoS capacity pool.

Delegate a subnet to Azure NetApp Files

You need to delegate a subnet to Azure NetApp Files so that Astra Control Service can create persistent volumes in that subnet. Note that Azure NetApp Files enables you to have only one delegated subnet in a VNet.

If you're using peered VNets, then both sides of the peering connection must be online: the VNet where your Kubernetes clusters reside and the VNet that has the Azure NetApp Files delegated subnet.

Step

1. Follow the Azure NetApp Files documentation to delegate a subnet to Azure NetApp Files.

After you're done

Wait about 10 minutes before discovering the cluster running in the delegated subnet.

Create an Azure service principal

Astra Control Service requires a Azure service principal that is assigned the Contributor role. Astra Control Service uses this service principal to facilitate Kubernetes application data management on your behalf.

A service principal is an identity created specifically for use with applications, services, and tools. Assigning a role to the service principal restricts access to specific Azure resources.

Follow the steps below to create a service principal using the Azure CLI. You'll need to save the output in a JSON file and provide it to Astra Control Service later on. Refer to Azure documentation for more details about using the CLI.

The following steps assume that you have permission to create a service principal and that you have the Microsoft Azure SDK (az command) installed on your machine.

Requirements

- The service principal must use regular authentication. Certificates aren't supported.
- The service principal must be granted Contributor or Owner access to your Azure subscription.
- The Azure subscription must contain the AKS clusters and your Azure NetApp Files account.

Steps

1. Identify the subscription and tenant ID where your AKS clusters reside (these are the clusters that you want to manage in Astra Control Service).

   ```
   az configure --list-defaults
   az account list --output table
   ```

2. Create the service principal, assign the Contributor role, and specify the scope to the entire subscription where the clusters reside.

   ```
   az ad sp create-for-rbac --name http://sp-astra-service-principal --role
   contributor --scopes /subscriptions/SUBSCRIPTION-ID
   ```

3. Store the resulting Azure CLI output as a JSON file.

   You'll need to provide this file so that Astra Control Service can discover your AKS clusters and manage Kubernetes data management operations. Learn about managing credentials in Astra Control Service.

4. Optional: Add the subscription ID to the JSON file so that Astra Control Service automatically populates the ID when you select the file.

   Otherwise, you'll need to enter the subscription ID in Astra Control Service when prompted.

   Example

```
{
  "appId": "0db3929a-bfb0-4c93-baee-aaf8",
  "displayName": "sp-example-dev-sandbox",
  "name": "http://sp-example-dev-sandbox",
  "password": "mypassword",
  "tenant": "011cdf6c-7512-4805-aaf8-7721afd8ca37",
  "subscriptionId": "99ce999a-8c99-99d9-a9d9-99cce99f99ad"
}
```

5. Optional: Test your service principal.

```
az login --service-principal --username APP-ID-SERVICEPRINCIPAL
--password PASSWORD --tenant TENANT-ID
az group list --subscription SUBSCRIPTION-ID
az aks list --subscription SUBSCRIPTION-ID
az storage container list --subscription SUBSCRIPTION-ID
```