Set up your cloud provider

Astra Control Service

NetApp
April 24, 2024

Table of Contents

Set up your cloud provider

Set up Amazon Web Services

A few steps are required to prepare your Amazon Web Services project before you can manage Amazon Elastic Kubernetes Service (EKS) clusters with Astra Control Service.

Quick start for setting up Amazon Web Services

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

[One] Review Astra Control Service requirements for Amazon Web Services

Ensure that clusters are healthy and running a supported version of Kubernetes, that worker nodes are online and running Linux or Windows, and more. Learn more about this step.

[Two] Create an Amazon account

If you don't already have an Amazon account, you need to create one so that you can use EKS. Learn more about this step.

[Three] Install the Amazon Web Services CLI

Install the AWS CLI so that you can manage AWS from the command line. Follow step-by-step instructions.

[Four] Optional: Create an IAM user

Create an Amazon Identity and Access Management (IAM) user. You can also skip this step and use an existing IAM user with Astra Control Service.

Read step-by-step instructions.

[Five] Create and attach a permissions policy

Create a policy with the required permissions for Astra Control Service to interact with your AWS account.

Read step-by-step instructions.

[Six] Save the credentials for the IAM user

Save the credentials for the IAM user so that you can import the credentials in to Astra Control Service.

Read step-by-step instructions.

EKS 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 1.25 to 1.28.

Image type

The image type for each worker node 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.

Astra Control Provisioner

Astra Control Provisioner and an external snapshot controller are required for operations with storage backends. To enable these operations, do the following:

1. Install the snapshot CRDs and the snapshot controller.
2. Enable Astra Control Provisioner.
3. Create a VolumeSnapshotClass.

CSI drivers for Amazon Elastic Block Store (EBS)

If you use the Amazon EBS storage backend, you need to install the Container Storage Interface (CSI) driver for EBS (it is not installed automatically).

Refer to the steps for instructions for installing the CSI driver.

Install an external snapshotter

If you haven't already done so, install the snapshot CRDs and the snapshot controller.

Install the CSI driver as an Amazon EKS add-on

1. Create the Amazon EBS CSI driver IAM role for service accounts. Follow the instructions in the Amazon documentation, using the AWS CLI commands in the instructions.
2. Add the Amazon EBS CSI add-on using the following AWS CLI command, replacing information in brackets <> with values specific to your environment. Replace <DRIVER_ROLE> with the name of the EBS CSI driver role that you created in the previous step:

```
aws eks create-addon \
  --cluster-name <CLUSTER_NAME> \
  --addon-name aws-ebs-csi-driver \
  --service-account-role-arn
arn:aws:iam::<ACCOUNT_ID>:role/<DRIVER_ROLE>
```

Configure the EBS storage class

1. Clone the Amazon EBS CSI driver GitHub repository to your system.

```
git clone https://github.com/kubernetes-sigs/aws-ebs-csi-
driver.git
```

2. Navigate to the dynamic-provisioning example directory.

```
cd aws-ebs-csi-driver/examples/kubernetes/dynamic-provisioning/
```

3. Deploy the ebs-sc storage class and ebs-claim persistent volume claim from the manifests directory.

```
kubectl apply -f manifests/storageclass.yaml
kubectl apply -f manifests/claim.yaml
```

4. Describe the ebs-sc storage class.

```
kubectl describe storageclass ebs-sc
```

   You should see output describing the storage class attributes.

Create an Amazon account

If you don’t already have an Amazon account, you need to create one to enable billing for Amazon EKS.

Steps

1. Go to the Amazon homepage , select **Sign in** at the top right, and select **Start here**.
2. Follow the prompts to create an account.

Install the Amazon Web Services CLI

Install the AWS CLI so that you can manage AWS resources from the command line.

Step

1. Go to Getting started with the AWS CLI and follow the instructions to install the CLI.

Optional: Create an IAM user

Create an IAM user so that you can use and manage AWS services and resources with increased security. You can also skip this step, and use an existing IAM user with Astra Control Service.

Step

1. Go to Creating IAM users and follow the instructions to create an IAM user.

Create and attach a permissions policy

Create a policy with the required permissions for Astra Control Service to interact with your AWS account.

Steps

1. Create a new file called `policy.json`.
2. Copy the following JSON content into the file:

```
{
    "Version": "2012-10-17",
    "Statement": [
        {
            "Sid": "VisualEditor0",
            "Effect": "Allow",
            "Action": [
                "cloudwatch:GetMetricData",
                "fsx:DescribeVolumes",
                "ec2:DescribeRegions",
                "s3:CreateBucket",
                "s3:ListBucket",
                "s3:PutObject",
                "s3:GetObject",
                "iam:SimulatePrincipalPolicy",
                "s3:ListAllMyBuckets",
                "eks:DescribeCluster",
                "eks:ListNodegroups",
                "eks:DescribeNodegroup",
                "eks:ListClusters",
                "iam:GetUser",
                "s3:DeleteObject",
                "s3:DeleteBucket",
                "autoscaling:DescribeAutoScalingGroups"
            ],
            "Resource": "*"
        }
    ]
}
```

3. Create the policy:

```
POLICY_ARN=$(aws iam create-policy  --policy-name <policy-name> --policy
-document file://policy.json  --query='Policy.Arn' --output=text)
```

4. Attach the policy to the IAM user. Replace `<IAM-USER-NAME>` with either the user name of the IAM user you created, or an existing IAM user:

```
aws iam attach-user-policy --user-name <IAM-USER-NAME> --policy-arn
=$POLICY_ARN
```

Save the credentials for the IAM user

Save the credentials for the IAM user so that you can make Astra Control Service aware of the user.

Steps

1. Download the credentials. Replace `<IAM-USER-NAME>` with the user name of the IAM user you want to use:

```
aws iam create-access-key --user-name <IAM-USER-NAME> --output json >
credential.json
```

Result

The `credential.json` file is created, and you can import the credentials in to Astra Control Service.

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.

> If you do not start out using Google Cloud Volumes Service for Google Cloud as a storage backend but plan to use it at a later date, you should complete the necessary steps to configure Google Cloud Volumes Service for Google Cloud now. Creating a service account later means that you might lose access to your existing storage buckets.

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 supported Kubernetes version, that worker nodes are online and running a supported image type, and more. Learn more about this step.

[Two] (Optional): Purchase Cloud Volumes Service for Google Cloud

If you plan to use Cloud Volumes Service for Google Cloud as a storage backend, go to the NetApp Cloud Volumes Service page in the Google Cloud Marketplace and select 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
  - Required for Cloud Volumes Service for Google Cloud
  - Optional (but recommended) for Google Persistent Disk
- 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
  - Required for Cloud Volumes Service for Google Cloud
  - Optional (but recommended) for Google Persistent Disk
- 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] (Optional): Set up network peering for your VPC

If you plan to use Cloud Volumes Service for Google Cloud as a storage backend, set up network peering from your VPC to Cloud Volumes Service for Google Cloud. Follow step-by-step instructions.

GKE cluster requirements

A Kubernetes cluster must meet the following requirements so you can discover and manage it from Astra Control Service. Note that some of these requirements only apply if you plan to use Cloud Volumes Service for Google Cloud as a storage backend.

Kubernetes version

A cluster must be running a Kubernetes version in the range of 1.26 to 1.28.

Image type

The image type for each worker node must be `COS_CONTAINERD`.

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

If you plan to use Cloud Volumes Service for Google Cloud as a storage backend, 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. As a best practice, you should choose a region that supports Cloud Volumes Service for Google Cloud, even if you do not use it as a storage backend. This makes it easier to use Cloud Volumes Service for Google Cloud as a storage backend in the future if your performance requirements change.

Networking

If you plan to use Cloud Volumes Service for Google Cloud as a storage backend, 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 address:

52.188.218.166/32

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.

Storage pools

If you use NetApp Cloud Volumes Service as a storage backend with the CVS service type, you need to configure storage pools before you can provision volumes. Refer to Service type, storage classes, and PV size for GKE clusters for more information.

Optional: Purchase Cloud Volumes Service for Google Cloud

Astra Control Service can use Cloud Volumes Service for Google Cloud as the storage backend for your persistent volumes. If you plan to use this service, 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, select **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 (Required for Cloud Volumes Service for Google Cloud)
- 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. Select **Account > Credentials > Add Credentials**, and then select **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. Select **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. Select **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)*

Optional: Set up network peering for your VPC

If you plan to use Cloud Volumes Service for Google Cloud as a storage backend service, 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 BlueXP 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.

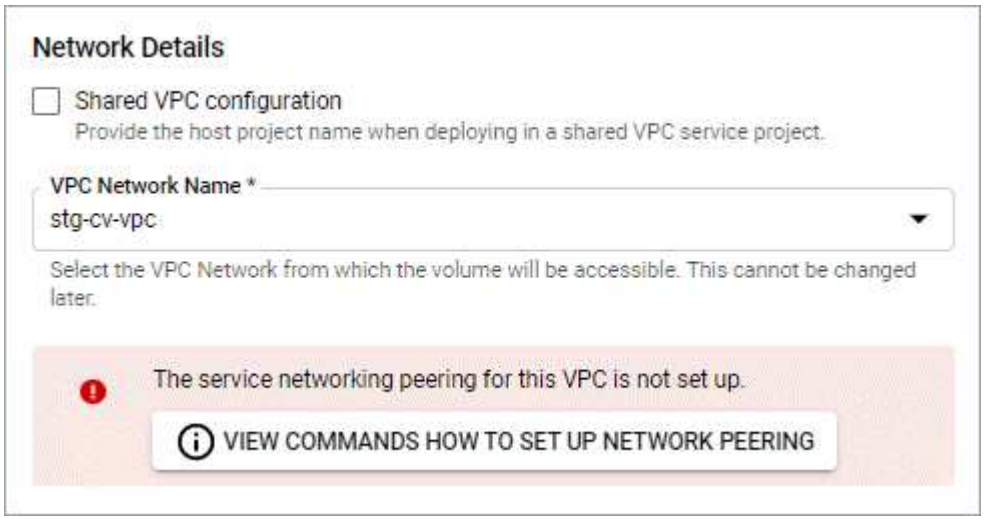
2. Go to Cloud Volumes in Google Cloud Platform.
3. On the **Volumes** page, select **Create**.
4. Under **Service Type**, select either **CVS** or **CVS-Performance**.

   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. Select 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 select cancel on the **Create File System** page.

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

Set up Microsoft Azure with Azure NetApp Files

A few steps are required to prepare your Microsoft Azure subscription before you can manage Azure Kubernetes Service clusters with Astra Control Service. Follow these instructions if you plan to use Azure NetApp Files as a storage backend.

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 a supported version of Kubernetes, that node pools are online and running Linux, and more. Learn more about this step.

[Two] Sign up for Microsoft Azure

Create a Microsoft Azure account. Learn more about this step.

[Three] Register for Azure NetApp Files

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

[Four] Create a NetApp account

Go to Azure NetApp Files in the Azure portal and create a NetApp account. Learn more about this step.

[Five] Set up capacity pools

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

[Six] 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.

[Seven] Create an Azure service principal

Create an Azure service principal that has the Contributor role. Learn more about this step.

[Eight] Optional: Configure redundancy for Azure backup buckets

By default, the buckets Astra Control Service uses to store Azure Kubernetes Service backups use the Locally Redundant Storage (LRS) redundancy option. As an optional step, you can configure a more durable level of redundancy for Azure buckets. Learn more about this step.

Azure Kubernetes Service 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 1.26 to 1.28.

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

Consider the following VNet requirements:

- 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 Amazon Kubernetes Service (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.

Sign up for Microsoft Azure

If you don't have a Microsoft Azure account, begin by signing up for Microsoft Azure.

Steps

1. Go to the Azure subscription page to subscribe to the Azure service.
2. Select a plan and follow the instructions to complete the subscription.

Register for Azure NetApp Files

Get access to Azure NetApp Files by registering the NetApp Resource Provider.

Steps

1. Log in to the Azure portal.
2. Follow Azure NetApp Files documentation to register the NetApp Resource Provider.

Create a NetApp account

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:

- The capacity pools need to be created in the same Azure region where the AKS clusters will be managed with Astra Control Service.
- 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 subscription or resource group you choose for scope 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. Do one of the following, depending on if you use an entire subscription or a resource group:
   - 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 service-principal-name --role
     contributor --scopes /subscriptions/SUBSCRIPTION-ID
     ```

   - Create the service principal, assign the Contributor role, and specify the resource group where the clusters reside.

     ```
     az ad sp create-for-rbac --name service-principal-name --role
     contributor --scopes /subscriptions/SUBSCRIPTION-
     ID/resourceGroups/RESOURCE-GROUP-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. Choose from the following example commands depending on the scope your service principal uses.

   Subscription scope

   ```
   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 --account-name STORAGE-ACCOUNT-NAME
   ```

   Resource group scope

   ```
   az login --service-principal --username APP-ID-SERVICEPRINCIPAL
   --password PASSWORD --tenant TENANT-ID
   az aks list --subscription SUBSCRIPTION-ID --resource-group RESOURCE-
   GROUP-ID
   ```

Optional: Configure redundancy for Azure backup buckets

You can configure a more durable redundancy level for Azure backup buckets. By default, the buckets Astra Control Service uses to store Azure Kubernetes Service backups use the Locally Redundant Storage (LRS) redundancy option. To use a more durable redundancy option for Azure buckets, you need to do the following:

Steps

1. Create an Azure storage account that uses the redundancy level you need using these instructions.
2. Create an Azure container in the new storage account using these instructions.
3. Add the container as a bucket to Astra Control Service. Refer to Add an additional bucket.

4. (Optional) To use the newly created bucket as the default bucket for Azure backups, set it as the default bucket for Azure. Refer to Change the default bucket.

Set up Microsoft Azure with Azure managed disks

A few steps are required to prepare your Microsoft Azure subscription before you can manage Azure Kubernetes Service clusters with Astra Control Service. Follow these instructions if you plan to use Azure managed disks as a storage backend.

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 a supported version of Kubernetes, that node pools are online and running Linux, and more. Learn more about this step.

[Two] Sign up for Microsoft Azure

Create a Microsoft Azure account. Learn more about this step.

[Three] Create an Azure service principal

Create an Azure service principal that has the Contributor role. Learn more about this step.

[Four] Configure Container Storage Interface (CSI) driver details

You need to configure your Azure subscription and the cluster to work with the CSI drivers. Learn more about this step.

[Five] Optional: Configure redundancy for Azure backup buckets

By default, the buckets Astra Control Service uses to store Azure Kubernetes Service backups use the Locally Redundant Storage (LRS) redundancy option. As an optional step, you can configure a more durable level of redundancy for Azure buckets. Learn more about this step.

Azure Kubernetes Service 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 1.26 to 1.28.

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

As a best practice, you should choose a region that supports Azure NetApp Files, even if you do not use it as a storage backend. This makes it easier to use Azure NetApp Files as a storage backend in the future if your performance requirements change. View Azure products by region.

CSI drivers

Clusters must have the appropriate CSI drivers installed.

Sign up for Microsoft Azure

If you don't have a Microsoft Azure account, begin by signing up for Microsoft Azure.

Steps

1. Go to the Azure subscription page to subscribe to the Azure service.
2. Select a plan and follow the instructions to complete the subscription.

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 subscription or resource group you choose for scope 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. Do one of the following, depending on if you use an entire subscription or a resource group:
   - 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 service-principal-name --role
contributor --scopes /subscriptions/SUBSCRIPTION-ID
```

- Create the service principal, assign the Contributor role, and specify the resource group where the clusters reside.

```
az ad sp create-for-rbac --name service-principal-name --role
contributor --scopes /subscriptions/SUBSCRIPTION-
ID/resourceGroups/RESOURCE-GROUP-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. Choose from the following example commands depending on the scope your service principal uses.

   Subscription scope

```
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 --account-name STORAGE-ACCOUNT-NAME
```

Resource group scope

```
az login --service-principal --username APP-ID-SERVICEPRINCIPAL
--password PASSWORD --tenant TENANT-ID
az aks list --subscription SUBSCRIPTION-ID --resource-group RESOURCE-
GROUP-ID
```

Configure Container Storage Interface (CSI) driver details

To use Azure managed disks with Astra Control Service, you'll need to install the required CSI drivers.

Enable the CSI driver feature in your Azure subscription

Before you install the CSI drivers, you need to enable the CSI driver feature in your Azure subscription.

Steps

1. Open the Azure command line interface.
2. Run the following command to register the driver:

```
az feature register --namespace "Microsoft.ContainerService" --name
"EnableAzureDiskFileCSIDriver"
```

3. Run the following command to ensure the change is propagated:

```
az provider register -n Microsoft.ContainerService
```

You should see output similar to the following:

```
{
"id": "/subscriptions/b200155f-001a-43be-87be-
3edde83acef4/providers/Microsoft.Features/providers/Microsoft.ContainerSer
vice/features/EnableAzureDiskFileCSIDriver",
"name": "Microsoft.ContainerService/EnableAzureDiskFileCSIDriver",
"properties": {
   "state": "Registering"
},
"type": "Microsoft.Features/providers/features"
}
```

Install the Azure managed disk CSI drivers in your Azure Kubernetes Service cluster

You can install the Azure CSI drivers to complete your preparation.

Step

1. Go to the Microsoft CSI driver documentation.
2. Follow the instructions to install the required CSI drivers.

Optional: Configure redundancy for Azure backup buckets

You can configure a more durable redundancy level for Azure backup buckets. By default, the buckets Astra Control Service uses to store Azure Kubernetes Service backups use the Locally Redundant Storage (LRS) redundancy option. To use a more durable redundancy option for Azure buckets, you need to do the following:

Steps

1. Create an Azure storage account that uses the redundancy level you need using these instructions.
2. Create an Azure container in the new storage account using these instructions.
3. Add the container as a bucket to Astra Control Service. Refer to Add an additional bucket.
4. (Optional) To use the newly created bucket as the default bucket for Azure backups, set it as the default bucket for Azure. Refer to Change the default bucket.

Copyright information

Trademark information