Astra Documentation for the Beta release

Project Astra

NetApp
November 17, 2020

This PDF was generated from https://docs.netapp.com/us-en/project-astra/index.html on November 17, 2020.
Always check docs.netapp.com for the latest.

Table of Contents

Astra Documentation for the Beta release

Get started

Intro to Astra

Astra is a Kubernetes application data lifecycle management service that simplifies operations for stateful applications. Easily back up Kubernetes apps, migrate data to a different cluster, and instantly create working application clones.

Features

The Astra beta program offers critical capabilities for Kubernetes application data lifecycle management:

- Create a protection policy for each of your apps
- Migrate applications and data from one Kubernetes cluster to another
- Easily clone an application from production to staging
- Create on-demand snapshots and backups
- Identify the health of your apps

Supported Kubernetes clusters

Astra can manage data for Google Kubernetes Engine (GKE) clusters.

On-prem Kubernetes clusters and clusters running in other cloud providers aren't supported at this time.

Learn more about cluster requirements.

How Astra works

Astra is a NetApp-managed cloud service that is always on and updated with the latest capabilities. It utilizes several components to enable application data lifecycle management. The following image shows the relationship between each component:

[diagram overview]

At a high level, Astra works like this:

- You set up your cloud provider.
- You add your first Kubernetes cluster to Astra. Astra then does the following:
  - Uses the cloud provider credentials that you provided to discover the cluster and the applications running on the cluster.

  - Creates an object store in your cloud provider account, which is where backup copies are sent.
  - Creates a new admin role on the cluster.
  - Uses the role to install NetApp's Trident, to create storage classes, and to eventually create namespaces and support cloning of applications.

    Astra uses Trident to provision persistent volumes backed by NetApp Cloud Volumes Service for Google Cloud.

    Astra creates three storage classes that use Cloud Volumes Service for Google Cloud: netapp-cvs-extreme, netapp-cvs-premium (default), and netapp-cvs-standard.

- At this point, cluster configuration is complete. You can now choose which apps to manage and start creating snapshots, backups, and clones.

Note that Astra continually watches your clusters for state changes, so it's aware of any new apps that you add along the way.

Join the Astra Beta release

There's still time to sign up for the Astra Beta program. Click this link and fill out the form to request participation. A NetApp representative will contact you soon after.

Release notes

Known issues

Known issues identify problems that might prevent you from using this release of the product successfully.

Clone performance impacted by large persistent volumes

Clones of very large and consumed persistent volumes might be intermittently slow, dependent on cluster access to the object store. If the clone is hung and no data has been copied for more than 30 minutes, Astra terminates the clone action.

Clone fails after deleting a backup

If you delete a backup that's used for an active clone operation, the clone fails and the newly cloned app gets stuck in a provisioning state. Contact NetApp support to clear the stuck app from the application listing.

Known limitations

Known limitations identify platforms, devices, or functions that are not supported by this release of the product, or that do not interoperate correctly with it. Review these limitations carefully.

One GCP project and one service account are supported

The Astra Beta program supports one Google Cloud Platform project and one service account. You should not add more than one service account to Astra and you shouldn't rotate service account credentials.

If you want to change the GCP project that you're using with Astra, then we will need to set up a new account for you.

We intend to address this limitation in a future release.

Persistent volume limit

You can have up to 100 volumes per Google Cloud region. If you reach this limit, creation of new clones or volumes will fail. Contact support to increase the volume limit.

Unhealthy pods affect app management

If a managed app has pods in an unhealthy state, Astra can't create new backups and clones.

Trident isn't uninstalled from a cluster

When you unmanage a cluster from Astra, Trident isn't automatically uninstalled from the cluster. To uninstall Trident, you'll need to follow these steps in the Trident documentation.

Existing connections to a Postgres pod causes failures

When you perform operations on Postgres pods, you shouldn't connect directly within the pod to use the psql command. Astra requires psql access to freeze and thaw the databases. If there is a pre-existing connection, the snapshot, backup, or clone will fail.

Get started with Astra

Quick start for Astra

This page provides a high-level overview of the steps that you need to complete to get started with Project Astra. The links within each step take you to a page that provides more details.

[Number 1] Review Kubernetes cluster requirements

- Astra supports Kubernetes clusters that are managed by Google Kubernetes Engine (GKE).

- Clusters must be running a healthy state, with at least one online worker node, in a supported Google Cloud region, and using CVS-Performance (not CVS Beta).
- A cluster must be running Kubernetes version 1.17 or later.
- The image type for each worker node must be Ubuntu.

Learn more about the Kubernetes cluster requirements.

[Number 2] Set up Google Cloud

a. Set up a Google Cloud account and project.
b. Create a service account that has the required permissions:
    - Kubernetes Engine Admin
    - NetApp Cloud Volumes Admin
    - Storage Admin
    - Service Usage Viewer
    - Compute Network Viewer
c. Create a service account key.
d. Enable the required APIs.
e. Enable networking for Cloud Volumes Service for Google Cloud.

Learn more about Google Cloud requirements.

[Number 3] Sign up to NetApp Cloud Central

Sign up to NetApp Cloud Central so you can access Astra and NetApp's other cloud services. Learn more about signing up.

[Number 4] Accept your Beta invitation

After you've been accepted into the Astra Beta program, you'll receive an invitation to join an Astra account. Accept this invitation to join the account and log in to the Astra interface. Learn more about accepting your invitation.

[Number 5] Add your first cluster

After you log in, click **Add a Kubernetes Cluster** to start managing your first cluster with Astra. Learn more about adding your first cluster.

Requirements

Get started by verifying support for your Kubernetes clusters, apps, and web browser.

Supported Kubernetes clusters

- The Astra beta program supports Kubernetes clusters that are managed by Google Kubernetes Engine (GKE).

  On-prem Kubernetes clusters and clusters running in other cloud providers are not supported at this time.

- Clusters must be running in a healthy state, in one of the supported regions:
  - us-central1
  - us-east4
  - us-west2
  - europe-west3
  - europe-west2
  - australia-southeast1
  - Other regions will be added in the future.
- A cluster must be running Kubernetes version 1.17 or later.
- The cluster must have at least one online worker node.
- The image type for each worker node must be Ubuntu.
- The cluster must be using CVS-Performance (not CVS Beta).
- If the cluster is private, it must have the master authorized network set to allow the Astra IP addresses:
  - 18.232.241.74
  - 52.72.239.237
  - 3.210.60.146

Supported apps

Astra supports all applications running on your Kubernetes clusters.

NetApp has validated some apps to ensure the safety and consistency of the snapshots and backups.

Learn the difference between a Validated and a Standard app.

No matter which type of app that you use with Astra, you should always test the backup and restore workflow yourself to ensure that you can meet your disaster recovery requirements.

Supported web browsers

Astra supports recent versions of Firefox, Safari, and Chrome with a minimum resolution of 1280 x 720.

Set up Google Cloud

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

Quick start for Google Cloud set up

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

[Number 1] Set up a Google Cloud account and project

You need a Google Cloud account and a project.

[Number 2] 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.

[Number 3] 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.

[Number 4] 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
- Service Consumer Management API

Follow step-by-step instructions.

[Number 5] Enable private services access

Set up private services access for 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.

[A conceptual diagram that shows a Google Cloud project]

Create a service account

Astra 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 user 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/project-astra/get-started/media/video-create-gcp-service-account.mp4

(video)

Create a service account key

Instead of providing a user name and password to Astra, you’ll provide a service account key when you add your first cluster. Astra 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/project-astra/get-started/media/video-create-gcp-service-account-key.mp4 *(video)*

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 last two APIs are required for Cloud Volumes Service for Google Cloud.

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

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

If you'd rather use the gcloud CLI, you can use these commands after setting your project:

```
gcloud services enable container.googleapis.com
gcloud services enable storage-component.googleapis.com
gcloud services enable storage-api.googleapis.com
gcloud services enable serviceusage.googleapis.com
gcloud services enable cloudresourcemanager.googleapis.com
gcloud services enable cloudvolumesgcp-api.netapp.com
gcloud services enable serviceconsumermanagement.googleapis.com
gcloud services enable servicenetworking.googleapis.com
gcloud services enable servicemanagement.googleapis.com
```

Enable private services access

Astra uses Cloud Volumes Service for Google Cloud as the backend storage for your persistent volumes. Other than the APIs that you enabled in the previous step, the only other requirement is to enable private services access to Cloud Volumes Service.

Step

1. Set up private services access from your project to create a high-throughput and low-latency data-path connection, as described in the Cloud Volumes Service for Google Cloud documentation.

Sign up to Cloud Central

The Astra beta program is integrated within NetApp Cloud Central's authentication service. Sign up to Cloud Central so you can access Astra and NetApp's other cloud services.

Steps

1. Open your web browser and go to NetApp Cloud Central.
2. In the top right, click **Sign up**.
3. Fill out the form and click **Sign up**.

   > You'll need to provide the email address that you enter in this form to the person who adds you to Astra.

   [A screenshot of the Cloud Central sign up page where you need to enter your email address, password, name, company, and your phone number, which is optional.]

4. Wait for an email from NetApp Cloud Central.
5. Click the link in the email to verify your email address.

Result

You now have an active Cloud Central user login.

Accept your Beta invitation

After you've been accepted into the Astra Beta program, you'll receive an invitation to join a Astra account. Accept this invitation to gain access to the Astra interface.

Steps

1. Open the email invitation to join a Astra account.

   [A screenshot of an email that invites you to join a Astra account. It includes a Join Now button that you can click to accept the invitation.]

2. Confirm that the email address in the invitation matches the email address that you used to sign up to Cloud Central.

If they don't match, then contact the person who added you to the account and let them know the email address that's associated with your Cloud Central account.

3. Click **Join Now**.

   A prompt should load in your web browser.

   [A screenshot that shows the Accept Invitation dialog box that appears in a web browser after you click the Join Now button from the email invitation.]

4. Click **Accept Invitation**.

   If you are the first person to join the Astra organization, you will be prompted to provide your address and serial number. **Be sure to use a valid physical address.** Please note the account name must be between 5 and 19 characters long. If you are being added to an existing account, you should now see the Astra interface.

   [A screenshot that shoes the Astra Dashboard.]

Add your first cluster to Astra

After you log in to the Astra beta program, your first step is to add a Kubernetes cluster.

Steps

1. On the Dashboard, click **Add a Kubernetes Cluster**.

   Follow the prompts to add the cluster.

2. **Provider**: Provide the service account key file either by uploading the file or by pasting the contents from your clipboard.

   [screenshot compute select credentials]

   Astra uses the service account to discover the clusters running in Google Kubernetes Engine.

3. **Compute**: Select the cluster that you'd like to add and click **Configure storage**.

   Pay careful attention to the Eligible tab. If a warning appears, hover over the warning to determine if there's an issue with the cluster. For example, it might identify the cluster doesn't have a worker node.

4. **Storage**: Select the default storage class that you'd like to use with this cluster and click **Review information**.

   Each storage class utilizes Cloud Volumes Service for Google Cloud.

5. **Review & Approve**: Review the configuration details and click **Add compute**.

   [screenshot compute approve]

The following video shows how to add a cluster.

▶ https://docs.netapp.com/us-en/project-astra/get-started/media/video-manage-cluster.mp4 *(video)*

Result

Astra creates an object store for application backups, creates an admin account on the cluster, and sets the default storage class that you specified. This process takes up to 5 minutes.

What’s next?

Now that you’ve logged in and added your first cluster to the Astra beta program, you’re ready to start using Astra’s application data management features.

- Start managing apps
- Protect apps
- Clone apps
- Invite and manage users
- Manage cloud provider credentials
- Manage notifications

Astra videos

Many of the pages on this doc site include videos that show you how to complete a task for Astra. If you’re just interested in videos, we’ve made it easy for you by collecting all of the videos on this single page (kind of like a playlist).

Videos for setting up Google Cloud

The following videos show how to complete set up requirements in Google Cloud before you can discover Kubernetes clusters running in GCP.

Create a service account

Astra uses a Google Cloud service account to facilitate Kubernetes application data management on your behalf. The following video shows how to create the service account from the Google Cloud console.

▶ https://docs.netapp.com/us-en/project-astra/get-started/media/video-create-gcp-service-account.mp4

(video)

Create a service account key

Astra uses a service account key to establish the identity of the service account that you just set up. The following video shows how to create the service account key from the Google Cloud console.

▶ https://docs.netapp.com/us-en/project-astra/get-started/media/video-create-gcp-service-account-key.mp4 *(video)*

Enable APIs

Your project needs permissions to access specific Google Cloud APIs. The following video shows how to enable the APIs from the Google Cloud console.

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

> 💡 Click here to view the full list of required APIs.

Videos for using Astra

The following videos show how to complete a few common tasks using Astra.

Add your first cluster to Astra

After you log in to Astra, your first step is to add a Kubernetes cluster.

▶ https://docs.netapp.com/us-en/project-astra/get-started/media/video-manage-cluster.mp4 *(video)*

Start managing an app

After you add a Kubernetes cluster to Astra, go to the Apps page to start managing the apps that run on the cluster.

▶ https://docs.netapp.com/us-en/project-astra/get-started/media/video-manage-app.mp4 *(video)*

Astra frequently asked questions for Beta

Overview

Welcome to the Astra Beta program!

Astra aims to simplify your application data lifecycle management operations for Kubernetes native applications. In the Beta, Astra support is limited to Kubernetes clusters running on Google Kubernetes Engine (GKE) on Google Compute Platform (GCP). Other cloud providers will be added in later phases of the project.

The following sections provide answers to some additional questions that you might come across as you use Astra. For any additional clarifications, please reach out to projectastra.feedback@netapp.com

Access to Astra

How can I access Astra?

Visit Astra at https://astra.netapp.io.

How can I get an invitation to the Beta?

Beta preview access is limited to a select few parties. Please register at https://cloud.netapp.com/project-astra-register.

I received an invitation to participate in the Beta. Where do I register my company?

Astra access is granted to your organization email address. This is the same email address that is registered with NetApp Cloud Central.

If you don't have a NetApp Cloud Central account yet, sign up using the **same** email in the invitation. You can create a NetApp Cloud Central account here: https://cloud.netapp.com.

I've added my colleagues to Astra, but they haven't received an email yet. What should I do?

Ask them to check their spam folder, or search their inbox for "invitation". You can also remove the user and attempt to re-add them. If neither of these work, please contact NetApp technical support with your organization name and the email addresses of people who haven't received the email invitation.

Registering Kubernetes Clusters

Can I add a private cluster to Astra?

Yes, you can add private clusters in Astra beta. To create a Google Kubernetes Engine (GKE) private cluster, follow the instructions in this knowledgebase article.

Private clusters must have the master authorized network set to allow the Astra IP addresses:

- 18.232.241.74
- 52.72.239.237
- 3.210.60.146

Can I use a custom network?

Yes, custom Virtual Private Cloud (VPC) networks are supported and Astra Beta will identify the right network peering and automate the required configuration.

Where can I find my service account credentials on GCP?

After you log in to the Google Cloud Console, your service account details will be in the **IAM and Admin** section. For more details, refer to how to set up Google Cloud for Astra.

How can I disable the service credentials I've registered with Astra?

When the Beta workflow testing is complete and you want to completely remove all credentials and objects from Astra, please contact NetApp Technical support and request to remove the Account. You can also invalidate any credentials stored with Astra by deleting the service account from the Google Cloud Console.

I’ve set permissions on the service account credentials in my GCP account, but it still doesn’t work. What should I do?

Contact NetApp Technical Support with a description of your problem and any error messages that you received.

I’ve changed my GCP service account roles. How do I update them in Astra?

Service account details are used when adding a GKE Kubernetes cluster to Astra. If the required roles and permissions are retained in the service account, you will not need to update anything in Astra.

If you rename or delete the service account, this will impact the application and cluster management features of Astra. You should contact projectastra.support@netapp.com to get help.

How many GCP service accounts can I register?

Different service accounts can be used when adding GKE clusters to Astra as long as they have the required roles and permissions. At a minimum, for each project, you need to provide one service account with the required roles and permissions.

How many Kubernetes clusters can I register?

You will need to register a minimum of two GKE Kubernetes clusters in order to exercise the Astra features. The maximum number of clusters for the Beta program is 100.

Do I need to install CSI drivers on my GKE cluster before adding it to Astra?

No. When your GKE cluster is added to Astra, the service will automatically install NetApp’s Trident Container Storage Interface (CSI) driver on the Kubernetes cluster. This CSI driver is used to provision persistent volumes backed by NetApp Cloud Volumes Service for Google Cloud.

I have a GKE cluster that’s running a different Kubernetes version than supported by Astra. Can I add that cluster to Astra?

The cluster discovery phase will not add a GKE cluster with an unsupported Kubernetes version. Astra provides information about supported Kubernetes version when it discovers a cluster running an unsupported Kubernetes version.

Can Astra validate the required GCP service account permissions?

Yes, Astra verifies that the required permissions are enabled before registering a GKE cluster, and will attempt to provide information about missing permissions.

How do I verify my GKE Kubernetes cluster is running supported Kubernetes version for Astra?

There are two ways you can verify the GKE Kubernetes cluster version:

1. Verify it from Google cloud console. Go to **Kubernetes Engine** > **Cluster** and select the relevant cluster. Check the Release Channel and Master Version.
2. Astra checks the GKE cluster version when the cluster is added. If Astra identifies an unsupported Kubernetes version, it provides more information in the Add compute user interface.

How do I know the worker nodes in the GKE Kubernetes cluster are running a supported image type?

The cluster discovery phase will not add a GKE cluster if the worker nodes are running an unsupported image type. If this happens, Astra will provide details on the supported image version (Ubuntu) in the Add compute user interface. Alternatively, you can verify the worker node image version from the Google Cloud Console.

How do I create a GKE cluster with a supported worker node image type?

When you create a GKE cluster or node pool, you can choose the operating system image that runs on each node. You can also upgrade an existing cluster to use a different node image type.

I would like to add different GKE clusters from different GCP projects. Is this supported in Astra?

No, this is not a supported configuration. Beta will only support a single GCP project.

How do I verify my GKE cluster was added successfully to Astra?

When you add the cluster, the user interface will show the status update and any error messages. When the cluster is added successfully, the status of the GKE cluster in the **Compute** section will be *Available*.

Alternatively, you can also verify if the Trident operator and CSI drivers deployed successfully under the namespace *trident* by running the kubectl commands:

`kubectl get pods -n trident`

or

`kubectl get pods -|grep trident`

I need to add worker nodes to my GKE cluster after adding to Astra. What should I do?

New worker nodes can be added to existing pools, or new pools can be created as long as they are the Ubuntu image type. These will be automatically discovered by Astra. If the new nodes are not visible in Astra, check if the new worker nodes are running the supported image type. You can also verify the health of the new worker nodes by using the `kubectl get nodes` command.

Can I unmanage my Kubernetes cluster from Astra?

Yes, you can remove one or more Kubernetes cluster from Astra at the same time. All managed applications from the unamanged cluster will be removed and Astra snapshots or backups taken of applications on that cluster will be unavailable to restore.

> Always remove a cluster from Astra before you delete it through GCP. Deleting a cluster from GCP while it's still being managed by Astra can cause problems for your Astra account.

What happens to my applications and data after removing the GKE cluster from Astra?

Removing a GKE cluster from Astra will not make any changes to the cluster's configuration (applications and persistent storage). Any Astra snapshots or backups taken of applications on that cluster will be unavailable to restore. Volume snapshot data stored within Cloud Volumes Service will not be removed. Persistent Storage backups created by Astra will remain within the Google Cloud object store, but they are unavailable for restore.

> Always remove a cluster from Astra before you delete it through GCP. Deleting a cluster from GCP while it's still being managed by Astra can cause problems for your Astra account.

Will NetApp Trident be uninstalled when I remove a GKE cluster from Astra?

Trident will not be uninstalled from a cluster when you remove it from Astra.

Managing Applications

How many apps per namespace?

There is no limitation about number applications under a namespace. Astra will discover all application in the name space by application name.

I have deployed my applications using Helm and kubectl. My newly-deployed application is not showing up on the Discovered Apps list. What can I check to identify the problem?

When an application is successfully deployed, Astra will automatically discover the application and add it to the Discovered Apps list. When applications are not listed in **Discovered Apps**, check the status and health of the Kubernetes pod by running `kubectl get pod -A |grep [pod name]`. If the pods are healthy and running, check to see if the application is listed under **Ignored Apps**.

I've deployed my applications using Helm and kubectl. I don't see any of my application's PVCs bound to GCP CVS. What's wrong?

The NetApp Trident operator sets the default storage class to `netapp-cvs-premium` after it's successfully added to Astra. When an application's PVCs are not bound to Cloud Volumes Services for Google Cloud, there are a few steps that you can take:

- Run `kubectl get sc` and check to see if the default storage class is set to *netapp-cvs*.
- Check the yaml file or Helm chart that was used to deploy the application and see if a different

storage class is defined.

- Check to make sure that the worker node image type is Ubuntu and the NFS mount succeeded.

I have an existing cluster that has applications using GCP persistent disks. Can I register those applications with Astra?

Applications using GCP PVCs will be discovered and registered by Astra. And it’s allowed to perform Astra data management operations. But snapshots and backups taken with Astra for those applications will not be application consistent.

How many applications can I simultaneously manage with Astra?

Multiple applications from different GKE cluster can be managed at the same time.

I moved my application to the Ignored list by mistake. Can I manage the applications that are on the Ignore list?

Yes, applications on the Ignored list can be registered successfully. Data management operations will function as usual after you start managing the application.

Can I register applications that are not MySQL, Jenkins, or PostgreSQL?

Yes, we can use data management services offered by Astra on any persistent volumes managed by Cloud Volumes Service for Google Cloud. However, application-level consistent snapshots, backup, migration, etc. will not be orchestrated through Astra.

Can Astra deploy an application?

Astra doesn’t deploy an application. Applications must be deployed outside of Astra by using kubectl or Helm charts.

What storage classes can I use in my PVCs to support Astra data management operations?

As part of adding the GKE cluster to Astra, NetApp Trident will create three different storage classes for Cloud Volume Services in GCP. Astra data management operations are only supported on storage class *netapp-cvs-extreme*, *netapp-cvs-premium*, and *netapp-cvs-standard*. And you can choose either of these storage class as default when adding a Kubernetes cluster to Astra.

What happens to applications after I stop managing them from Astra?

Applications, data, and any existing backups or snapshots remain available. Data management operations will not be available for unmanaged applications or any backups or snapshots that belong to it. When the application is managed by Astra again, the existing snapshots and backups will be available for data management operations.

Data Management Operations

My application uses several PVs. Will Astra take snapshots and backups of all these PVCs?

Astra aims to simplify application data lifecycle management. Using Astra eliminates the need for individual volume-level data management operations. A snapshot operation on an application by Astra includes snapshot of all the PVs that are bound to the application’s PVCs.

Can I create snapshot schedules and assign retention schedules?

Yes, you can use the Configure Protection Policy option to set a retention policy for each individual application.

What is the difference between snapshots and backups?

Snapshot refers to local snapshots, where data is stored as part of the provisioned volumes. Given that they are stored on the same provisioned volume, they are usually faster. Local snapshots are used to restore the application to an earlier point in time.

Backups are stored on object storage. They could be slower compared to the local snapshots. However, they can be accessed across regions in the cloud. Backups are used for migrating applications across regions in the cloud. Also, a user can choose to have longer retention period for backups.

Can I manage snapshots taken by Astra directly through the Cloud Volumes Service snapshot management interface or object storage?

Snapshots and backups taken through Astra can only be managed through Astra. Astra provides interfaces to create, view, and delete the snapshots and backups. If data objects associated with these snapshots are managed outside of the Astra interface, it can result in intermittent behavior.

Use Project Astra

Manage and protect apps

Start managing apps

After you add a Kubernetes cluster to the Astra beta program, go to the Apps page to start managing the apps that run on the cluster.

Start managing an app

View the apps that you can discover from the **Discovered** section of the Apps page and then click **Manage**.

Steps

1. Click **Apps** and then click **Discovered**.

   If you just added the cluster to Astra, you'll notice that some apps are in the process of being discovered.

   [screenshot app discovery]

   If there are any issues with discovery, you can hover over the icon in the Ready column to view details about the issue.

   In the following image, you can see that Astra is still in the process of discovering the app. Hovering over the Ready column shows the current status.

   [screenshot app discovery status]

   After Astra discovers an app, you have the option to either manage the app or ignore it.

2. Look at the **Group** column to see which namespace the application is running in (it's designated with the folder icon) and whether any Kubernetes labels are available (those are designated with a tag icon).

   Here's an example:

   [screenshot group]

   This information can be helpful because you might want to manage everything in the namespace, or you might want to manage the app using labels that you've already set up. You'll see how to use these labels in a few steps.

3. Click the drop-down list in the **Actions** column for the desired app and click **Manage**.

[screenshot app manage]

4. In the **Manage Application** dialog box, provide the required information to manage the app:
   a. **New App**: Customize the name of the app.
   b. **Selected Resources**: View and manage the selected Kubernetes resources that you'd like to protect (pods, secrets, persistent volumes, and more). Here's an example:

      [screenshot selected resources]

      There are two primary ways to use the Selected Resources field:

      - View the resources to validate that the Kubernetes resources that you want to protect are listed.
      - If a namespace contains multiple discrete applications and you use Kubernetes labels to split apart the apps, then you can choose a label to register the app with, based on that label.
        - View the available labels by expanding a resource and clicking the number of labels.

          [screenshot view labels]

        - Select one of the labels.

          [screenshot select label]

          After you choose a label, it displays in the **Label** field. Astra also updates the **Unselected Resources** section to show the resources that don't match the selected label.

        - View **Unselected Resources** to verify the app resources that you don't want to protect.

          [screenshot selected label]

5. Click **Manage App**.

The following video shows how to start managing an app.

▶ https://docs.netapp.com/us-en/project-astra/use/media/video-manage-app.mp4 *(video)*

Result

Astra enables management of the app. You can now find it in the **Managed** tab.

[screenshot app managed]

What's next?

Repeat these steps for additional apps. Choose **Ignore** for any of the apps that you don't want to manage from Astra. Those apps will move to the **Ignored** tab. Ideally, you'd have zero clusters listed in the Discovered tab after you're done.

Manage an app using a custom label

Astra includes an action at the top of the Apps page named **Manage new app**. You can use this action to manage an app by using a *custom* label. For example, you might not want to use one of the discovered Helm labels to manage the app.

Steps

1. Click **Apps** > **Manage new app**.
2. In the **Manage Application** dialog box, provide the required information to manage the app:
   a. **New App**: Customize the name of the app.
   b. **Compute**: Select the compute where the app resides.
   c. **Namespace:** Select the namespace for the app.
   d. **Label:** Enter a custom label.
   e. **Selected Resources:** View and manage the Kubernetes resources that you'd like to protect.
   f. **Unselected Resources**: Verify the app resources that you don't want to protect.
3. Click **Manage App**.

Result

Astra enables management of the app. You can now find it in the **Managed** tab.

What about system apps?

When you add a Kubernetes cluster, Astra also discovers the system apps running on the cluster. You can view them by filtering the Apps list.

[screenshot system apps]

We don't show you these system apps by default because it's rare that you'd need to back them up.

Protect apps with snapshots and backups

Protect your apps by taking snapshots and backups using an automated protection policy or on an ad-hoc basis.

Snapshots and backups

A *snapshot* is a point-in-time copy of an app that's stored on the same provisioned volume as the app. They are usually fast. Local snapshots are used to restore the application to an earlier point in time.

A *backup* is stored on object storage. A backup can be slower to take compared to the local snapshots. However, they can be accessed across regions in the cloud. Backups are used for migrating applications across cloud regions. Also, you can choose to have a longer retention period for backups.

Configure a protection policy

Configure a protection policy to protect an app by creating snapshots, backups, or both at a defined schedule and with a specified number of copies to retain.

Steps

1. Click **Apps** and then click the name of an app.
2. Click **Data Protection**.
3. Click **Configure Protection Policy**.

   [A screenshot of the Data protection tab for an app which enables you to configure a protection policy.]

4. Define a protection schedule by choosing the number of snapshots and backups to keep hourly, daily, weekly, and monthly.

   You can define the hourly, daily, weekly, and monthly schedules concurrently. A schedule won't turn active until you set a retention level.

   The following example sets a schedule to take snapshots daily and weekly, while retaining the last 14 hourly snapshots and the last 26 weekly snapshots. It also takes monthly backups and retains the last 12 copies. Because 0 copies were selected for hourly, no hourly snapshots or backups are taken.

   [A screenshot of a sample configuration policy where you can choose to take snapshots and backups on an hourly, daily, weekly, or monthly basis.]

5. Click **Review Information**.
6. Click **Set Protection Policy.**

Result

Astra implements the data protection policy by creating and retaining snapshots and backups using the schedule and retention policy that you defined.

Create a snapshot

You can create an on-demand snapshot at any time.

Steps

1. Click **Apps**.
2. Click the drop-down list in the **Actions** column for the desired app.
3. Click **Snapshot**.

   [A screenshot of the app page where you can click the drop-down list in the actions column and select Snapshot.]

4. Customize the name of the snapshot and then click **Review Information**.
5. Review the snapshot summary and click **Snapshot App**.

Result

Astra creates a snapshot of the apps.

Create a backup

You can also back up an app at any time.

Steps

1. Click **Apps**.
2. Click the drop-down list in the **Actions** column for the desired app.
3. Click **Backup**.

   [A screenshot of the app page where you can click the drop-down list in the actions column and select Backup.]

4. Customize the name of the backup, choose whether to back up the app from an existing snapshot, and then click **Review Information**.
5. Review the backup summary and click **Backup App**.

Result

Astra creates a backup of the app.

View snapshots and backups

You can view the snapshots and backups of an app from the Data Protection tab.

Steps

1. Click **Apps** and then click the name of an app.
2. Click **Data Protection**.

   The snapshots display by default.

   [A screenshot of the data protection tab for an app where you can view the list of the current snapshots and backups.]

3. Click **Backups** to see the list of backups.

Delete snapshots

Delete the scheduled or on-demand snapshots that you no longer need.

Steps

1. Click **Apps** and then click the name of an app.
2. Click **Data Protection**.
3. Click the drop-down list in the **Actions** column for the desired snapshot.
4. Click **Delete**.

   [A screenshot of the Data protection tab for an app where you can delete a snapshot.]

5. Type the name of the snapshot to confirm deletion and then click **Yes, Delete snapshot**.

Result

Astra deletes the snapshot.

Delete backups

Delete the scheduled or on-demand backups that you no longer need.

1. Click **Apps** and then click the name of an app.
2. Click **Data Protection**.
3. Click **Backups**.

   [A screenshot of the Backups option that's available in the far right of the data protection tab.]

4. Click the drop-down list in the **Actions** column for the desired backup.
5. Click **Delete**.

   [A screenshot of the Data protection tab for an app where you can delete a snapshot.]

6. Type the name of the backup to confirm deletion and then click **Yes, Delete backup**.

Result

Astra deletes the backup.

Restore apps

You can restore an app by creating a clone from a point-in-time snapshot or from a backup.

Steps

1. Click **Apps**.
2. Click the drop-down list in the **Action** column for the desired app.
3. Click **Clone**.

   [A screenshot of the app page where you can click the drop-down list in the actions column and select Clone.]

4. **Clone details**: Specify details for the clone:
   - Enter a name.
   - Choose whether to restore the app to the same cluster or to a different cluster.
   - Choose to create the clone from an existing snapshot or backup.
5. **Source**: Choose the snapshot or backup that you'd like to use.

   [screenshot clone source]
6. **Clone Summary**: Review the details about the clone and click **Clone App**.

   [screenshot clone summary]

Result

Astra restores the app based on the information that you provided.

Clone and migrate apps

Clone an existing app to create a duplicate app on the same Kubernetes cluster or on another cluster. Cloning can help if you need to move applications and storage from one Kubernetes cluster to another. For example, you might want to move workloads through a CI/CD pipeline and across Kubernetes namespaces.

When Astra clones an app, it creates a clone of your application configuration and persistent storage.

Steps

1. Click **Apps**.
2. Click the drop-down list in the **Action** column for the desired app.
3. Click **Clone**.

   [A screenshot of the app page where you can click the drop-down list in the actions column and select Clone.]
4. **Clone details**: Specify details for the clone:
   - Enter a name.
   - Choose a destination cluster for the clone.
   - Choose whether you want to create the clone from an existing snapshot or backup. If you don't select this option, Astra creates the clone from the app's current state.
5. **Source**: If you chose to clone from an existing snapshot or backup, choose the snapshot or backup that you'd like to use.

[screenshot clone source]

6. **Clone Summary**: Review the details about the clone and click **Clone App**.

   [screenshot clone summary]

Result

Astra clones that app based on the information that you provided.

View app and cluster health

View a summary of app and cluster health

Click the **Dashboard** to see a high-level view of your apps, clusters, and their health.

[screenshot dashboard]

The Apps tile helps you identify the following:

- How many apps you're currently managing with Astra.
- Whether those managed apps are healthy.
- Whether the apps are fully protected (they're protected if recent backups are available).
- The number of apps that were discovered, but are not yet managed.

  Ideally, this number would be zero because you would either manage or ignore apps after they're discovered. And then you would monitor the number of discovered apps on the Dashboard to identify when developers add new apps to a cluster.

Note that these aren't just numbers or statuses—you can drill down from each of these. For example, if you have an unhealthy app, you can hover over the icon to identify which app has an issue and what the issue is. You can then go to the cluster to correct the issue.

[screenshot dashboard healthy]

The Compute tile provides similar details about the health of your clusters and you can drill down to get more details just like you can with an app.

View the health and details of a cluster

After you add a cluster to the Astra beta program, you can view details about the cluster, such as its location, the worker nodes, persistent volumes, and storage classes.

Steps

1. Click **Compute**.
2. Click the name of a cluster.
3. View the information in the **Overview** and **Storage** tabs to find the information that you're looking for.
   - **Overview**: Details about the worker nodes, including their state.
   - **Storage**: The persistent volumes associated with the cluster, including the storage class and state.

     [A screenshot of the Overview tab for a cluster.]

View the health and details of an app

After you start managing an app, Astra provides details about the app that enables you to identify its status (whether it's healthy), its protection status (whether it's fully protected in case of failure), the pods, persistent storage, and more.

[screenshot app overview]

Steps

1. Click **Apps** and then click the name of an app.
2. Click around to find the information that you're looking for:

   App Status

   Provides a status that reflects the app's state in Kubernetes. For example, are pods and persistent volumes online? If an app is unhealthy, you'll need to go and troubleshoot the issue on the cluster by looking at Kubernetes logs. Astra doesn't provide information to help you fix a broken app.

   App Protection Status

   Provides a status of how well the app is protected. An app is either Fully Protected (it has a recent backup), Partially Protected (it has a data protection schedule or active snapshots, but no recently successful backup), or it's Unprotected.

   You can't be fully protected until you have a recent backup. This is important because backups are stored in an object store away from the persistent volumes. If a failure or accident wipes out the cluster and it's persistent storage, then you need a backup to recover.

   Overview

   Information about the state of the pods that are associated with the app.

Data protection

Enables you to configure a data protection policy and to view the existing snapshots and backups.

Storage

Shows you the app-level persistent volumes. The state of a persistent volume is from the perspective of the Kubernetes cluster.

Resources

Enables you to verify which resources are being backed up and managed.

Manage your account

Invite and remove users

Invite users to join your Astra beta program account and remove users that should no longer have access to your account.

Invite users

Account Owners and Admins can invite other users to join the Astra account.

Steps

1. Click **Account**.
2. In the **Users** tab, click + **Invite users**.
3. Enter the user’s name, email address, and their role.

   Note the following:

   - The email address must match the email address that the user used to sign up to Cloud Central.
   - Each role provides the following permissions:
     - An **Owner** has Admin permissions and can delete accounts.
     - An **Admin** has Member permissions and can invite other users.
     - A **Member** can fully manage apps and clusters.
     - A **Viewer** can view resources.

       [A screenshot of the Invite Users screen where you enter a name]
4. Click **Send invite/s**.

Result

The user will receive an email that invites them to join your account.

Remove users

An Account Owner can remove other users from the account at any time.

Steps

1. Click **Account**.
2. In the **Users** tab, select the users that you want to remove.
3. Click **Actions** and select **Remove user/s**.
4. When you’re prompted, confirm deletion by typing the user’s name and then click **Yes, Remove User**.

Result

Astra removes the user from the account.

Add and remove credentials

Add and remove cloud provider credentials from your account at any time. The Astra beta program uses these credentials to discover clusters, apps on a cluster, and to provision resources on your behalf.

Note that all users in Astra share the same sets of credentials.

Add credentials

The most common way to add credentials to Astra is when you add a cluster, but you can also add credentials from the Account page. The credentials will then be available to choose when you add your next Kubernetes cluster.

What you’ll need

You should have the service account key file for a service account that has the required permissions. Learn more.

Steps

1. Click **Account** > **Credentials**.
2. Click **Add Credentials**.
3. Enter a name for the credentials that distinguishes them from other credentials in Astra.
4. Provide the Google Cloud service account key file either by uploading the file or by pasting the contents from your clipboard.
5. Click **Add credentials**.

Result

The credentials are now available to select when you add a Kubernetes cluster to Astra.

Remove credentials

An Account Owner can remove credentials from the account at any time.

Steps

1. Click **Account** > **Credentials**.
2. Click the drop-down list in the **State** column for the credentials that you want to remove.
3. Click **Remove**.

   [A screenshot of the Credentials tab in the Account page where you can click the state column and select the Remove action.]

4. Type the name of the credentials to confirm deletion and then click **Yes, Remove Credentials**.

Result

Astra removes the credentials from the account.

View and manage notifications

Astra notifies you when actions have completed or failed. For example, you’ll see a notification if a backup of an app completed successfully.

The number of unread notifications is available in the top right of the interface:

[A screenshot that shows the Astra interface where you can view the number of unread notifications.]

You can view these notifications and mark them as read (this can come in handy if you like to clear unread notifications like we do).

Steps

1. Click the number of unread notifications in the top right.

   [A screenshot that shows the expanded notifications in the Astra interface.]

2. Review the notifications and then click **Mark as read** or **Show all notifications**.

   If you clicked **Show all notifications**, the Notifications page loads.

3. On the **Notifications** page, view the notifications, select the ones that you want to mark as read, click **Action** and select **Mark as read**.

Unmanage apps and clusters

Remove any apps or clusters that you no longer want to manage from the Astra beta program.

Stop managing an app

Stop managing apps that you no longer want to back up, snapshot, or clone from Astra.

About this task

- This action stops your app from being managed by Astra. It doesn't remove the app from your cluster.
- Existing backups and snapshots aren't deleted, but data management operations aren't available from Astra.
- If you remove an app from your cluster, Astra will identify that it was removed, but the app remains in a managed state. That means you can still clone the app from a backup until you explicitly unmanage the app.

> ⚠ Always remove a cluster from Astra before you delete it through GCP. Deleting a cluster from GCP while it's still being managed by Astra can cause problems for your Astra account.

Steps

1. Click **Apps**.
2. Click the checkbox for the apps that you no longer want to manage.
3. Click the **Actions** drop-down and select **Unmanage application/s**.
4. Confirm that you want to unmanage the apps and then click **Yes, Unmanage Apps**.

Result

Astra stops managing the app.

Stop managing clusters

Stop managing the clusters that you no longer want to manage from Astra.

About this task

- This action stops your cluster from being managed by Astra. It doesn't make any changes to the cluster's configuration and it doesn't delete the cluster.

  Trident won't be uninstalled from the cluster. Learn how to uninstall Trident.
- Apps associated with this cluster will no longer be managed and will go into a Detached state.
- Any snapshots or backups associated with applications on the cluster aren't deleted, but data management operations aren't available from Astra.

Steps

1. Delete any snapshots or backups and stop managing the cluster's applications before removing the cluster.

Learn how to delete snapshots and backups.

2. Click **Compute**.
3. Click the checkbox for the clusters that you no longer want to manage.
4. Click the **Actions** drop-down and select **Unmanage cluster/s**.
5. Confirm that you want to unmanage the clusters and then click **Yes, Unmanage Compute**.

Result

Astra stops managing the clusters.

Cleaning up after the Beta

A few things remain after you've removed apps and clusters from Astra:

- The object store that was created in your Google Cloud account
- The Admin role that was installed on each managed Kubernetes cluster
- The service account that you created

You'll need to manually remove these.

Learn

Validated vs Standard Apps

There are two types of applications you can bring to Astra: Validated and Standard. Learn the difference between these two categories, and the potential impacts on your projects and strategy.

> It’s tempting to think of these two categories as "supported" and "unsupported." But as you will see, there is no such thing as an "unsupported" app in Astra. You can add any app to Astra, although validated apps have more infrastructure built around their Astra workflows compared to standard apps.

Validated Apps

Validated apps for the Astra Beta Program are:

- MySQL 0.3.22
- MariaDB 14.14
- PostgreSQL 11.7
- Jenkins 2.249.1 LTS

The short list of validated apps represents applications that Astra recognizes. The Astra QA team has analyzed and confirmed these apps to be fully tested to restore.

Validated apps have also been checked by the Astra Development team, which creates custom workflows to help ensure the safety and consistency of your data. For example, when Astra takes a backup of a PostgreSQL database, it first quiesces the database. After the backup is complete, Astra restores the database to normal operation.

No matter which type of app you use with Astra, always test the backup and restore workflow yourself to ensure that you can meet your disaster recovery requirements.

Let us know what apps you would like to see validated in the future. Contact us through the Feedback email address on the Support page.

Standard Apps

Any other app, including custom programs, is considered a standard app. You can add and manage standard apps through Astra. You can also create basic crash-consistent Snapshots and Backups of a standard app. However, these have not been QA-tested to restore the app to its original state.

Define a custom app

Creating a custom app lets you group elements of your Kubernetes cluster into a single app.

A custom app gives you more granular control over what to include in a Astra operation, including:

- Clone
- Snapshot
- Backup
- Protection Policy

In most cases you will want to use Astra's features on your entire app. However, you can also create a custom app to use these features by the labels you assign to Kubernetes objects in a namespace.

To create a custom app, go to the Apps page and click + **Define custom app**.

As you make your selections, the Custom App window will show you which resources will be included or excluded from your custom app. This helps you make sure you are choosing the correct criteria for defining your custom app.

[A screenshot of the Define Custom App pop-up window showing which resources will and won't be included in the custom app]

In the above example, one resource (the pod `nginx-pod0` labeled `deployment:canary`) will be included in the custom app. Two pods (`nginx-pod` and `nginx-pod2` both labeled `deployment:stable`) will be excluded.

> ℹ Custom apps can only be created within a specified namespace on a single cluster. The Astra beta program does not support the ability for a custom app to span multiple namespaces or clusters.

A label is a key/value pair you can assign to Kubernetes objects for identification. Labels make it easier to sort, organize, and find your Kubernetes objects. To learn more about Kubernetes labels, see the official Kubernetes documentation.

> ⚠ Overlapping policies for the same resource under different names can cause data conflicts. If you create a custom app for a resource, be sure it's not being cloned or backed up under any other policies.

Example: Separate Protection Policy for canary release

In this example, the devops team is managing a canary release deployment. Their cluster has three pods running NginX. Two of the pods are dedicated to the stable release. The third pod is for the canary release.

The devops team’s Kubernetes admin adds the label `deployment=stable` to the stable release pods. She adds the label `deployment=canary` to the canary release pod.

[A screenshot of the result of running 'kubectl get pods --namespace=nginx-app --show-labels' showing the Kubernetes labels on the three pods.]

The team’s stable release includes a requirement for hourly snapshots and daily backups. The canary release is more ephemeral, so they want to create a less aggressive, short-term Protection Policy for anything labeled `deployment=canary`.

In order to avoid possible data conflicts, the admin will create two custom apps: one for the canary release, and one for the stable release. This keeps the backups, snapshots, and clone operations separate for the two groups of Kubernetes objects.

After she adds the cluster to the Astra beta program, her next step is to define a custom app. To do this, she clicks the + **Define custom app** button on the Apps page.

In the pop-up window which appears, she sets `devops-canary-deployment` as the app name. She chooses the cluster in the **Compute** drop-down, then the app’s namespace from the **Namespace** drop-down.

At this point, she can either type `deployment=canary` in the **Labels** field, or select that label from the resources listed below.

After defining the custom app for the canary deployment, she repeats the process for the stable deployment.

When she has finished creating the two custom apps, she can treat these resources as any other Astra application. She can clone them, create backups and snapshots, and create a custom Protection Policy for each group of resources based on her Kubernetes labels.

Solutions

MySQL/MariaDB

Deploy MariaDB From a Helm Chart

Learn how to exercise the Astra Beta program workflow by deploying MariaDB from a Helm chart. After you deploy MariaDB on your cluster, you can manage the application with Astra.

MariaDB is a validated app for the Astra Beta program. Learn the difference between Validated and Standard apps.

> ℹ The Astra Beta program only supports MySQL 0.3.22 and MariaDB 14.14.

System Requirements

In order to deploy MariaDB from a Helm chart for the Astra Beta program, you need the following:

- A GKE cluster which has been added to Astra.
- Updated versions of Helm (version 3.2+) and Kubectl installed.
- Kubeconfig configured using the gcloud tool with a command like `gcloud container clusters get-credentials my-cluster-name`

Install MariaDB

To exercise the Astra Beta program workflow, we recommend you use the Helm chart.

Deploy MariaDB with the command:

```
helm install mariadb bitnami/mariadb --namespace testdb --create-namespace --set
db.database=test_db,db.user=test_db_user,db.password=NKhjs2wQPt8 > /dev/null 2>&1
```

This does the following:

- Creates the `testdb` namespace.
- Deploys MariaDB on the `testdb` namespace.
- Creates a database named `test_db`
- Creates a user `test_db_user` with password `NKhjs2wQPt8`

> ⚠ This method of setting the password at deployment is insecure. Only use this command when setting up MariaDB for a sandbox deployment to use Astra Beta program. We do not recommend this for a production environment.

After the Helm chart is deployed, it will be automatically discovered by Astra, at which point you can manage the app with Astra.

Deploy MySQL From a Helm Chart

Learn how to exercise the Astra Beta workflow by deploying MySQL from a Helm chart. After you deploy MySQL on your Kubernetes cluster, you can manage the application with Astra.

MariaDB and MySQL are validated apps for the Astra Beta program. Learn the difference between Validated and Standard apps.

> ℹ The Astra beta program only supports MySQL 0.3.22 and MariaDB 14.14.

System Requirements

In order to deploy MySQL from a Helm chart for the Astra Beta program, you need the following:

- A GKE cluster which has been added to Astra.
- Updated versions of Helm (version 3.2+) and Kubectl installed.
- Kubeconfig configured using the gcloud tool with a command like `gcloud container clusters get-credentials my-cluster-name`

> ℹ You must deploy your app after the cluster is added to Astra, not before.

Install MySQL

To exercise the Astra Beta workflow, we recommend the standard stable chart.

> ℹ You must deploy the Helm chart in a namespace other than the default.

Deploy MySQL with the command:

```
helm install mysql stable/mysql --namespace testdb--set
db.database=test_db,db.user=test_db_user,db.password=NKhjs2wQPt8
if you need to deploy mysql under a new namespace; please use the following command
helm install mysql stable/mysql --namespace testdb --create-namespace --set
db.database=test_db,db.user=test_db_user,db.password=NKhjs2wQPt8
```

This does the following:

- Creates the `testdb` namespace.
- Deploys MySQL on the `testdb` namespace.
- Creates a database named `test_db`

- Creates a user `test_db_user` with password `NKhjs2wQPt8`

> This method of setting the password at deployment is insecure. You can use own secrets and config maps and passing along with helm command.

After the Helm chart is deployed, it will be automatically discovered by Astra, at which point you can manage the app with Astra.

Work With MySQL/MariaDB on Astra

This guide focuses on Helm as the preferred way to deploy Postgres apps. Plain YAML and Operator-based deployments may be covered in future guides.

For express instructions on launching MySQL/MariaDB on Astra, see Deploy MySQL/MariaDB from a Helm Chart.

MariaDB and MySQL are validated apps for the Astra Beta program. Learn the difference between Validated and Standard apps

> MySQL 0.3.22 and MariaDB 14.14 are the only versions supported in the Astra Beta program.

Requirements

In order to deploy MySQL/MariaDB from a Helm chart on a cluster registered with the Astra Beta program, you will need the following:

GKE Cluster

An up-to-date Kubernetes cluster (version 1.17+) which is connected to Astra. For help creating your GKE cluster and connecting it to Astra, see the Getting Started Guide.

Kubectl

Kubectl is a standard tool for interacting with Kubernetes. For more information, see the guide Install and Set Up kubectl in the official Kubernetes documentation.

Kubeconfig

The Kubeconfig file contains the credentials which let kubectl communicate with your Kubernetes cluster. to learn how to download your GKE Kubeconfig file, see the Google Cloud guide for configuring cluster access for kubectl.

Cloud Volume Serivice in Google Cloud Platform (CVS-GCP)

CVS is the storage layer and connective elements for Astra, respectively. More details on how to configure CVS on GCP may be found in the workflow guide for CVS^.

Helm (v3)

Helm is a popular way to organize and install apps on Kubernetes. To install Helm on your local computer, follow their handy install guide.

MySQL/MariaDB Requirements

For a MySQL/MariaDB application, Astra requires:

- `global.storageClass` value to be set to the storageClass representing either CVS or Trident (or, that storageClass is set as your cluster's default provisioner).

Install MariaDB/MySQL

For the Astra beta, we recommend the custom Helm chart we have created for this purpose. For instructions on how to deploy from this custom chart, see Deploy MySQL/MariaDB from a Helm Chart.

The values need to be set to consume the volumes provisioned by CVS, be deployed in a namespace other than default, and your stateful app needs to be available to Astra.

By default the Bitnami chart uses a cluster's default storage class. Kubernetes clusters registered with Astra beta use Trident CSI fron NetApp. Trident automatically sets CVS as the default storage class. Use `kubectl get sc` to see what your cluster's storageClasses are. This produces output like the following:

```
NAME                           PROVISIONER            RECLAIMPOLICY   VOLUMEBINDINGMODE
ALLOWVOLUMEEXPANSION   AGE
netapp-cvs-extreme             csi.trident.netapp.io   Delete          Immediate
true                   26h
netapp-cvs-premium (default)   csi.trident.netapp.io   Delete          Immediate
true                   26h
netapp-cvs-standard            csi.trident.netapp.io   Delete          Immediate
true                   26h
standard                       kubernetes.io/gce-pd    Delete          Immediate
true                   27h
```

You have two options for changing settings in your `values.yaml`. The first option is to open the file and edit it directly. The second option is to add an extra argument to your usual Helm CLI command.

To view and export `values.yaml`, use the `helm show` command:

```
# mariaDB
helm show values bitnami/mariadb
# mySQL
helm show values bitnami/mysql
```

or

```
# mariaDB
helm show values bitnami/mariadb > my-values.yaml
# mySQL
helm show values bitnami/mysql > my-values.yaml
```

This creates a `my-values.yaml` file in your local directory. That file is a copy of the official `values.yaml`.

Dry Run

Before deploying, you can do a dry run to make sure everything is set up correctly.

To do this, edit the values in the `my-values.yaml` file you created in the previous step. Test your deployment using the `-f my-values.yaml` and `--dry-run` flags:

```
# MariaDB
helm install -f my-values.yaml --namespace testdb --generate-name bitnami/mariadb --dry
-run

# MySQL
helm install -f my-values.yaml --namespace testdb --generate-name bitnami/mysql --dry-run
```

If the output from our dry run looks correct, we may deploy to your cluster by removing `--dry-run`.

Before we can run the helm charts for real, you can choose to use an existing namespace or specify to create a new namespace with helm command like below;

```
# MariaDB
helm install -f my-values.yaml --namespace testdb --generate-name bitnami/mariadb
--create-namespace

# MySQL
 helm install -f my-values.yaml --namespace testdb --generate-name bitnami/mysql --create
-namespace
```

After deploying the application using Helm chart Astra will be automatically discover the application. After a successful discovery you can manage the app with Astra.

Postgres

Deploy Postgres From a Helm Chart

Learn how to exercise the Astra beta program workflow by deploying Postgres from a Helm chart. After you deploy Postgres on your cluster, you can register the application with Astra.

Postgres is a validated app for the Astra Beta program. Learn the difference between Validated and Standard apps.

> The Astra Beta Program only supports Postgres 11.7.

System Requirements

In order to deploy Postgres from a Helm chart for the Astra alpha program, you need the following:

- A fresh GKE cluster which has been added to Astra.
- Updated versions of Helm (version 3.2+) and Kubectl installed.
- Kubeconfig configured using the gcloud tool with a command like `gcloud container clusters get-credentials my-cluster-name`

Namespace Requirements

You must deploy your app in a namespace other than the default. In the following example, we create and use the namespace `testdb` for the deployment.

A namespace which is empty for more than 60 seconds will be ignored by Astra. Thus, you want to be sure to deploy your app into your namespace within one minute after you create the namespace.

In the following example, we use `&&` to concatenate the commands for creating the namespace and deploying the app. We recommend this approach, as it ensures the commands are run in sequence even if you get interrupted.

We recommend the use of `&&` instead of `;` to concatenate commands. `&&` is conditional, and only runs the second command if the first command completes successfully.

> You must deploy your app after the cluster is added to Astra, not before.

Install Postgres

To exercise the Astra alpha workflow, we recommend the standard stable chart.

> You must deploy the Helm chart in a namespace other than the default.

Deploy Postgres with the command:

```
kubectl create namespace testdb && helm install stable/postgresql --namespace testdb
--set postgresqlPassword=U9dH9HT4pWS,postgresqlDatabase=test_db --generate-name
```

This does the following:

- Creates the `testdb` namespace.

- Deploys Postgres on the `testdb` namespace.
- Creates a database named `test_db`
- Creates a user `test_db_user` with password `U9dH9HT4pWS`

> This method of setting the password at deployment is insecure. Only use this command when setting up Postgres for a sandbox deployment to use Astra alpha program. We do not recommend this for a production environment.

After the Helm chart is deployed, it will be automatically detected by Astra, at which point you can register the app with Astra. Please note that for the Astra alpha program, it can take up to 5 minutes for applications to show up in the Discovered Applications list after being installed.

Work With Postgres on Astra

This guide focuses on Helm as the preferred way to deploy Postgres apps. Plain YAML and Operator-based deployments may be covered in future guides.

For express instructions on launching Postgres on Astra, see Deploy Postgres from a Helm Chart.

Postgres is a validated app for the Astra Beta program. Learn the difference between Validated and Standard apps.

> Postgres 11.7 is the only version supported in the Astra beta program.

Requirements

In order to deploy Postgres from a Helm chart on a cluster registered with Astra, you will need the following:

GKE Cluster

An up-to-date Kubernetes cluster (version 1.17+) which is connected to Astra. For help creating your GKE cluster and connecting it to Astra, see the Getting Started Guide.

Kubectl

Kubectl is a standard tool for interacting with Kubernetes. For more information, see the guide Install and Set Up kubectl in the official Kubernetes documentation.

Kubeconfig

The Kubeconfig file contains the credentials which let kubectl communicate with your Kubernetes cluster. to learn how to download your GKE Kubeconfig file, see the Google Cloud guide for configuring cluster access for kubectl.

CVS and Cloud Central

CVS and Cloud Central are the storage layer and connective elements for Astra, respectively. More details on how to configure CVS on GCP may be found in the workflow guide for CVS.

Helm (v3)

Helm is a popular way to organize and install apps on Kubernetes. To install Helm on your local computer, follow their handy install guide.

Postgres Requirements

For a Posgres application, Astra Alpha requires:

- `global.storageClass` value to be set to the storageClass representing either CVS or Trident (or, that storageClass is set as your cluster's default provisioner).
- The namespace set to something other than default, using the `--namespace` argument.
- A single node deployment. Multi-node and HA deployments will be supported in future releases.

The Astra alpha program does not support replicas or failovers. Only single instance versions of the databases are supported. For testing Astra in Alpha, leave replication off, and check that the `global.storageClass` value in `values.yaml` is pointing to the correct storageClass.

Namespace Requirements

You must deploy your app in a namespace other than the default. In the following example, we create and use the namespace `testdb` for the deployment.

A namespace which is empty for more than 60 seconds will be ignored by Astra. Thus, you want to be sure to deploy your app into your namespace within one minute after you create the namespace.

In the following example, we use `&&` to concatenate the commands for creating the namespace and deploying the app. We recommend this approach, as it ensures the commands are run in sequence even if you get interrupted.

We recommend the use of `&&` instead of `;` to concatenate commands. `&&` is conditional, and only runs the second command if the first command completes successfully.

Using psql on Astra

During the Astra alpha program, if you need to perform operations on Postgres pods (such as creating or restoring from backup), be sure to exit out of the psql client if you are using it on the pod.

Astra requires psql access to freeze and thaw the databases. If there is a pre-existing connection the snapshot/backup/clone operation will fail.

Install Postgres

For the Astra alpha program, we recommend the Bitnami Postgres chart. To install this chart, see

Deploy Postgres from a Helm Chart.

The values need to be set to consume the volumes provisioned by CVS, be deployed in a namespace other than default, and your stateful app needs to be available to Astra.

By default the Bitnami Postgres chart uses a cluster's default dynamic provisioner. Since Trident (part of Astra) automatically sets CVS as the default storage class, you should be in good shape. Use `kubectl get sc` to see what your cluster's storageClasses are. This produces output like the following:

```
NAME                           PROVISIONER            RECLAIMPOLICY   VOLUMEBINDINGMODE
ALLOWVOLUMEEXPANSION   AGE
netapp-cvs-extreme             csi.trident.netapp.io   Delete          Immediate
true                   26h
netapp-cvs-premium (default)   csi.trident.netapp.io   Delete          Immediate
true                   26h
netapp-cvs-standard            csi.trident.netapp.io   Delete          Immediate
true                   26h
standard                       kubernetes.io/gce-pd    Delete          Immediate
true                   27h
```

You have two options for changing settings in your `values.yaml`. The first option is to open the file and edit it directly. The second option is to add an extra argument to your usual Helm CLI command.

To view and export `values.yaml`, use the `helm show` command:

```
helm show values bitnami/postgresql
```

or

```
helm show values bitnami/postgresql > my-values.yaml
```

This creates a `my-values.yaml` file in your local directory. That file is a copy of the official `values.yaml`.

Dry Run

Before deploying, you can do a dry run to make sure everything is set up correctly.

To do this, edit the values in the `my-values.yaml` file you created in the previous step. Test your deployment using the `-f my-values.yaml` and `--dry-run` flags:

```
helm install -f my-values.yaml --namespace testdb --generate-name bitnami/postgresql
--dry-run
```

If the output from our dry run looks correct, we may deploy to your cluster by removing `--dry-run`.

Before we can run the helm charts for real, we need to first create the namespace. We've chosen `testdb` and may use kubectl to create that namespace.

```
kubectl create namespace testdb && helm install -f my-values.yaml --namespace testdb
--generate-name bitnami/postgresql
```

After the Helm chart is deployed, it will be automatically detected by Astra, at which point you can register the app with Astra. Please note that for the Astra alpha program, installed applications can take up to 5 minutes to show up in the Discovered Applications list.

Generate test data

Helm provides instructions for connecting to newly-installed Postgres apps. These instructions should contain a few different methods for connecting to the database.

This process is also discussed here in the Postgres documentation.

```
NOTES:
** Please be patient while the chart is being deployed **
PostgreSQL can be accessed via port 5432 on the following DNS name from within your
cluster:
    postgresql-1591290927.longship.svc.cluster.local - Read/Write connection
To get the password for "postgres" run:
    export POSTGRES_PASSWORD=$(kubectl get secret --namespace longship postgresql-
1591290927 -o jsonpath="{.data.postgresql-password}" | base64 --decode)
To connect to your database run the following command:
    kubectl run postgresql-1591290927-client --rm --tty -i --restart='Never' --namespace
longship --image docker.io/bitnami/postgresql:11.8.0-debian-10-r19
--env="PGPASSWORD=$POSTGRES_PASSWORD" --command -- psql --host postgresql-1591290927 -U
postgres -d postgres -p 5432
To connect to your database from outside the cluster execute the following commands:
    kubectl port-forward --namespace longship svc/postgresql-1591290927 5432:5432 &
    PGPASSWORD="$POSTGRES_PASSWORD" psql --host 127.0.0.1 -U postgres -d postgres -p 5432
```

From your own instructions, copy the line below `To get the password for "postgres" run:` and run it. Next, copy the lines below `To connect to your database run the following command:` and run them.

This will put you in the psql command line tool. Using psql, you may generate test data for testing Astra snapshot, clone, and restore features.

An example chunk of SQL that generates 10,000 rows is included in this guide.

```
-- create a db
CREATE DATABASE astra_test_db;
-- connect to it
\c astra_test_db;
-- create a table
CREATE TABLE junk(
  id      SERIAL PRIMARY KEY,
  title   VARCHAR(32) NOT NULL UNIQUE
);
-- insert 10,000 rows into the table
INSERT INTO junk (
    title
)
SELECT md5(i::text)
FROM generate_series(1, 10000) g_s(i);
-- check that data looks correct
SELECT * FROM junk LIMIT 20;
```

Jenkins

Deploy Jenkins From a Helm Chart

Learn how to exercise the Astra beta program workflow by deploying Jenkins from a Helm chart. After you deploy Jenkins on your cluster, you can register the application with Astra.

Jenkins is a validated app for the Astra Beta program. Learn the difference between Validated and Standard apps.

Requirements

The following requirements are necessary for installing and running Jenkins on a Kubernetes cluster for the Astra beta program.

Compatibility Requirements

Only the current version of Jenkins (5.0.26) has been officially validated for use with Astra. Other versions may work, but may only run as a standard application.

Astra does not support the Kubernetes plugin for Jenkins at this time. This functionality will be added soon. You can run Jenkins in a Kubernetes cluster without the plugin. The plugin provides scalability to your Jenkins cluster.

System Requirements

- A new Kubernetes cluster which has been added to Astra.

- Helm (version 3.2+) and kubectl installed on your local computer.
- Kubeconfig configured using the gcloud tool with a command like `gcloud container clusters get-credentials my-cluster-name`

Namespace Requirements

You must deploy your app in a namespace other than the default. In the following example, we create and use the namespace `jenkins` for the deployment.

In the following example, we use `&&` to concatenate the commands for creating the namespace and deploying the app. We recommend this approach, as it ensures the commands are run in sequence even if you get interrupted.

We recommend the use of `&&` instead of `;` to concatenate commands. `&&` is conditional, and only runs the second command if the first command completes successfully.

> ℹ You must deploy your app after the cluster is added to Astra, not before.

Install Jenkins

To exercise the Astra beta workflow, we recommend you use the Bitnami Helm chart. Add the Bitnami chart repo:

```
helm repo add bitnami https://charts.bitnami.com/bitnami
```

> ℹ You must deploy the Helm chart in a namespace other than the default.

Create the `jenkins` namespace and deploy Jenkins into it with the command:

```
kubectl create namespace jenkins && helm install jenkins --namespace jenkins --set
persistence.storageClass=netapp-cvs-perf-premium,persistence.size=100Gi bitnami/jenkins
```

This does the following:

- Creates the `jenkins` namespace.
- Sets the correct storage class.
- Sets the persistent volume storage size to 100Gi.

After the Helm chart is deployed, it will be automatically detected by Astra. You can then register the app with Astra.

Knowledge and support

Register for support

Astra attempts to automatically register your account for support when you set up your account. If it can't, then you can manually register for support yourself. Support registration is required to obtain help from NetApp technical support.

Verify your support registration

Astra includes a Support Status field that enables you to confirm your support registration.

Steps

1. Click **Support**.
2. Take a look at the Support Status field.

   The Support Status starts off as "Not Registered" but then moves to "In-Progress" and finally to "Registered" once complete.

   You can refresh your screen to get a current snapshot of process.

   If you have any issues registering your serial number, contact us at projectastra.feedback@netapp.com.

Obtain your serial number

When you accept your Beta invitation, Astra prompts you to set up your account. Astra uses the information that you provide about your company to generate a 20-digit NetApp serial number that starts with "941".

The NetApp serial number represents your Astra account. You'll need to use this serial number when opening a web ticket.

You can find your serial number in the Astra interface from the **Support** page.

[screenshot support]

Activate support entitlement

If Astra was unable to automatically register your account for support, then you must register the NetApp serial number associated with Astra to activate support entitlement. We offer 2 options for support registration:

1. Current NetApp customer with existing NetApp Support Site (NSS) SSO account

2. New NetApp customer with no existing NetApp Support Site (NSS) SSO account

Option 1: Current NetApp customer with an existing NetApp Support Site (NSS) account

Steps

1. Navigate to the Cloud Data Services Support Registration page to create an NSS account.
2. Click **I am already registered as a NetApp customer**.
3. Enter your NetApp Support Site credentials to log in.

   The Existing Customer Registration page displays.

   [Existing Customer Registration Form]

4. Complete the required information on the form:
   a. Enter your name, company, and email address.
   b. Select **Astra** as the product line.
   c. Enter your serial number.
   d. Click **Submit Registration**.

Result

You should be redirected to a "Registration Submitted Successfully" page. The email address associated with your registration will receive an email within a couple minutes stating that "your product is now eligible for support."

This is a one-time support registration for the applicable serial number.

Option 2: New NetApp customer with no existing NetApp Support Site (NSS) account

Steps

1. Navigate to the Cloud Data Services Support Registration page to create an NSS account.
2. Click **I am not a registered NetApp Customer**.

   The New Customer Registration page displays.

   [New Customer Registration Form]

3. Complete the required information on the form:
   a. Enter your name and company information.
   b. Select **Astra** as the Product Line.
   c. Enter your serial number.
   d. Click **Submit Registration**.

You will receive a confirmation email from your submitted registration. If no errors occur, you will be re-directed to a "Registration Submitted Successfully" page. You will also receive an email within an hour stating that "your product is now eligible for support".

This is a one-time support registration for the applicable serial number.

4. As a new NetApp customer, you also need to create a NetApp Support Site (NSS) user account for future support activations and for access to the support portal for technical support chat and web ticketing.

   Go to the NetApp Support Registration site to perform this task. You can provide your newly registered Astra serial number to expedite the process.

Get help

NetApp provides support for the Astra beta program in a variety of ways. Extensive free self-support options are available 24x7, such as knowledgebase (KB) articles and a Slack channel. Your Astra account includes remote technical support via web ticketing.

You must first activate support for your NetApp serial number in order to use these non self-service support options. A NetApp Support Site (NSS) SSO account is required for chat and web ticketing along with case management.

You can access support options from the Astra UI by selecting the **Support** tab from the main menu.

[screenshot support]

Self support

These options are available for free 24x7:

- Knowledge base

  Search for articles, FAQ’s, or Break Fix information related to Astra.

- Documentation

  This is the doc site that you’re currently viewing.

- Slack

  Go to the containers channel in thePub workspace to connect with peers and experts.

- Feedback email

  Send an email to projectastra.feedback@netapp.com to let us know your thoughts, ideas, or concerns.

Subscription support

In addition to the self-support options above, you can work with a NetApp Support Engineer to resolve any issues after you activate support for your NetApp serial number.

Once your Astra serial number is activated, you can access NetApp technical support resources by creating a Support ticket.

Select **Cloud Data Services** > **Astra**.

Use your "941" serial number to open the web ticket. Learn more about your serial number.

[screenshot web ticket]

Legal notices

Legal notices provide access to copyright statements, trademarks, patents, and more.

Copyright

http://www.netapp.com/us/legal/copyright.aspx

Trademarks

NETAPP, the NETAPP logo, and the marks listed on the NetApp Trademarks page are trademarks of NetApp, Inc. Other company and product names may be trademarks of their respective owners.

http://www.netapp.com/us/legal/netapptmlist.aspx

Patents

A current list of NetApp owned patents can be found at:

https://www.netapp.com/us/media/patents-page.pdf

Privacy policy

https://www.netapp.com/us/legal/privacypolicy/index.aspx

Open source

Notice files provide information about third-party copyright and licenses used in NetApp software.

Notice for Astra

Copyright Information

Copyright © 2020 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.

RESTRICTED RIGHTS LEGEND: Use, duplication, or disclosure by the government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause at DFARS 252.277-7103 (October 1988) and FAR 52-227-19 (June 1987).

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.