Deploy apps

Astra Control Center

NetApp
June 17, 2022

Table of Contents

Deploy apps

Deploy Jenkins from a Helm chart

Learn how to deploy Jenkins from the Bitnami Helm chart. After you deploy Jenkins on your cluster, you can register the application with Astra Control.

Jenkins is a validated app for Astra Control.

- Learn the difference between a validated app and a standard app in Astra Control Service.
- Learn the difference between a validated app and a standard app in Astra Control Center.

These instructions apply to both Astra Control Service and Astra Control Center.

> Applications deployed from Google Marketplace have not been validated. Some users report issues with discovery and/or backup with Google Marketplace deployments of Postgres, MariaDB, and MySQL.

Requirements

- A cluster that has been added to Astra Control.

  > For Astra Control Center, you can add the cluster to Astra Control Center first or add the app first.

- Updated versions of Helm (version 3.2+) and Kubectl installed on a local machine with the proper kubeconfig for the cluster

Astra Control does not currently support the Kubernetes plugin for Jenkins. You can run Jenkins in a Kubernetes cluster without the plugin. The plugin provides scalability to your Jenkins cluster.

Install Jenkins

Two important notes on this process:

- You must deploy your app after the cluster is added to Astra Control Service, not before. Astra Control Center will accept applications before or after the cluster is added to Astra Control Center.
- You must deploy the Helm chart in a namespace other than the default.

Steps

1. Add the Bitnami chart repo:

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

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

```
helm install <name> bitnami/jenkins -namespace <namespace> --create
-namespace -set global.storageClass=<storage_class_name>
```

If the volume size is changed, use Kibibyte (Ki), Mebibyte (Mi) or Gibibyte (Gi) units.

You need to define the storage class only in these situations:

- You are using Astra Control Service and you don't want to use the default storage class.
- You are using Astra Control Center and haven't yet imported the cluster into Astra Control Center. Or, you have imported the cluster, but don't want to use the default storage class.

Result

This does the following:

- Creates a namespace.
- Sets the correct storage class.

After the pods are online, you can manage the app with Astra Control. Astra Control enables you to manage an app at the namespace level or by using a helm label.

Deploy MariaDB from a Helm chart

Learn how to deploy MariaDB from the Bitnami Helm chart. After you deploy MariaDB on your cluster, you can manage the application with Astra Control.

MariaDB is a validated app for Astra.

- Learn the difference between a validated app and a standard app in Astra Control Service.
- Learn the difference between a validated app and a standard app in Astra Control Center.

These instructions apply to both Astra Control Service and Astra Control Center.

Applications deployed from Google Marketplace have not been validated. Some users report issues with discovery and/or backup with Google Marketplace deployments of Postgres, MariaDB, and MySQL.

Requirements

- A cluster that has been added to Astra Control.

For Astra Control Center, you can add the cluster to Astra Control Center first or add the app first.

- Updated versions of Helm (version 3.2+) and Kubectl installed on a local machine with the proper kubeconfig for the cluster

Install MariaDB

Two important notes on this process:

- You must deploy your app after the cluster is added to Astra Control Service, not before. Astra Control Center will accept applications before or after the cluster is added to Astra Control Center.
- You must deploy the Helm chart in a namespace other than the default.

Steps

1. Add the Bitnami chart repo:

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

2. Deploy MariaDB with the command:

```
helm install <name> bitnami/MariaDB -namespace <namespace> --create
-namespace -set global.storageClass=<storage_class_name>
```

   If the volume size is changed, use Kibibyte (Ki), Mebibyte (Mi) or Gibibyte (Gi) units.

   You need to define the storage class only in these situations:

   - You are using Astra Control Service and you don't want to use the default storage class.
   - You are using Astra Control Center and haven't yet imported the cluster into Astra Control Center. Or, you have imported the cluster, but don't want to use the default storage class.

Result

This does the following:

- Creates a namespace.
- Deploys MariaDB on the namespace.
- Creates a database.

This method of setting the password at deployment is insecure. We do not recommend this for a production environment.

After the pods are online, you can manage the app with Astra Control. Astra Control enables you to manage an app at the namespace level or by using a helm label.

Deploy MySQL from a Helm chart

Learn how to deploy MySQL from the Bitnami Helm chart. After you deploy MySQL on your Kubernetes cluster, you can manage the application with Astra Control.

MySQL is a validated app for Astra Control.

- Learn the difference between a validated app and a standard app in Astra Control Service.

- Learn the difference between a validated app and a standard app in Astra Control Center.

These instructions apply to both Astra Control Service and Astra Control Center.

> Applications deployed from Google Marketplace have not been validated. Some users report issues with discovery and/or backup with Google Marketplace deployments of Postgres, MariaDB, and MySQL.

Requirements

- A cluster that has been added to Astra Control.

> For Astra Control Center, you can add the cluster to Astra Control Center first or add the app first.

- Updated versions of Helm (version 3.2+) and Kubectl installed on a local machine with the proper kubeconfig for the cluster

Install MySQL

Two important notes on this process:

- You must deploy your app after the cluster is added to Astra Control Service, not before. Astra Control Center will accept applications before or after the cluster is added to Astra Control Center.
- We recommend that you deploy the Helm chart in a namespace other than the default.

Steps

1. Add the Bitnami chart repo:

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

2. Deploy MySQL with the command:

```
helm install <name> bitnami/mysql -namespace <namespace>
--create-namespace -set global.storageClass=<storage_class_name>
```

> If the volume size is changed, use Kibibyte (Ki), Mebibyte (Mi) or Gibibyte (Gi) units.

You need to define the storage class only in these situations:

- You are using Astra Control Service and you don't want to use the default storage class.
- You are using Astra Control Center and haven't yet imported the cluster into Astra Control Center. Or, you have imported the cluster, but don't want to use the default storage class.

Result

This does the following:

- Creates a namespace.
- Deploys MySQL on the namespace.

After the pods are online, you can manage the app with Astra Control. Astra Control allows you to manage an app with its name, at the namespace level, or by using a helm label.

Deploy Postgres from a Helm chart

Learn how to deploy Postgres from the Bitnami Helm chart. After you deploy Postgres on your cluster, you can register the application with Astra Control.

Postgres is a validated app for Astra.

- Learn the difference between a validated app and a standard app in Astra Control Service.
- Learn the difference between a validated app and a standard app in Astra Control Center.

These instructions apply to both Astra Control Service and Astra Control Center.

> Applications deployed from Google Marketplace have not been validated. Some users report issues with discovery and/or backup with Google Marketplace deployments of Postgres, MariaDB, and MySQL.

Requirements

- A cluster that has been added to Astra Control.

  > For Astra Control Center, you can add the cluster to Astra Control Center first or add the app first.

- Updated versions of Helm (version 3.2+) and Kubectl installed on a local machine with the proper kubeconfig for the cluster

Install Postgres

Two important notes on this process:

- You must deploy your app after the cluster is added to Astra Control Service, not before. Astra Control Center will accept applications before or after the cluster is added to Astra Control Center.
- You must deploy the Helm chart in a namespace other than the default.

Steps

1. Add the Bitnami chart repo:

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

2. Deploy Postgres with the command:

```
helm install <name> bitnami/postgresql -namespace <namespace>
--create-namespace -set global.storageClass=<storage_class_name>
```

If the volume size is changed, use Kibibyte (Ki), Mebibyte (Mi) or Gibibyte (Gi) units.

You need to define the storage class only in these situations:

- You are using Astra Control Service and you don’t want to use the default storage class.
- You are using Astra Control Center and haven’t yet imported the cluster into Astra Control Center. Or, you have imported the cluster, but don’t want to use the default storage class.

Result

This does the following:

- Creates a namespace.
- Deploys Postgres on the namespace.

After the pods are online, you can manage the app with Astra Control. Astra Control enables you to manage an app at the namespace level or by using a helm label.