# Oracle® Communications

# Operations Services Overlay Installation and Upgrade Guide

Release 24.1.0

F94156-01

April 2024

**ORACLE®**

Oracle Communications Operations Services Overlay Installation and Upgrade Guide, Release 24.1.0

F94156-01

# Contents

# My Oracle Support

My Oracle Support (https://support.oracle.com) is your initial point of contact for all product support and training needs. A representative at Customer Access Support can assist you with My Oracle Support registration.

Call the Customer Access Support main number at 1-800-223-1711 (toll-free in the US), or call the Oracle Support hotline for your local country from the list at http://www.oracle.com/us/support/contact/index.html. When calling, make the selections in the sequence shown below on the Support telephone menu:

1. Select **2** for New Service Request.

2. Select **3** for Hardware, Networking and Solaris Operating System Support.

3. Select one of the following options:

   - For Technical issues such as creating a new Service Request (SR), select **1**.

   - For Non-technical issues such as registration or assistance with My Oracle Support, select **2**.

You are connected to a live agent who can assist you with My Oracle Support registration and opening a support ticket.

My Oracle Support is available 24 hours a day, 7 days a week, 365 days a year.

# Acronyms

The following table lists the acronyms and the terminologies used in the document:

**Table    Acronyms and Terminologies**

| Acronym | Description |
| --- | --- |
| alm | AlertManager |
| cm-r | Configmap-reload |
| CNC Console | Cloud Native Configuration Console |
| CNE | Oracle Communications Cloud Native Core, Cloud Native Environment |
| CSAR | Cloud Service Archive |
| GUI | Graphical User Interface |
| ONAP | Open Network Automation Platform |
| OSO | Operations Services Overlay |
| OSDC | Oracle Software Download Center |
| prom | Prometheus |
| svr | Server |

# What's New in This Guide

This section introduces the documentation updates for release 24.1.x.

**Release 24.1.0- F94156-01, April 2024**

- Updated the versions of the following software in the [Prerequisites](#) section.

    – Alertmanager version to 0.26.0

    – Prometheus version to 2.50.1

    – configmapreload to 0.12.0

    – nginx controller is replaced with 24_1_common_pod

- Updated the note about OSO upgrade in the [Upgrading OSO](#) section.

- Added a step to view the current version of OSO in the [Verifying Installation](#) section.

- Updated the examples for the following parameters in the [OSO Configuration Parameters](#) section.

    – `ALERTMANAGER-REPO-ADDRESS`

    – `PROMETHEUS-REPO-ADDRESS`

    – `HELM-TEST-IMAGE-REPO-ADDRESS`

    – `CONFIGMAP-RELOAD-REPO-ADDRESS`

- Added the following sections for rollback support in OSO:

    – [Supported Rollback Paths](#)

    – [Rollback Tasks](#)

    – [Rolling Back OSO](#)

- Updated the release version to 24.1.0 across the guide.

- Updated the upgrade path in the [Supported Upgrade Paths](#) section.

# 1
# Introduction

This guide describes how to install or upgrade Oracle Communications Operations Services Overlay (OSO) in a cloud native environment.

> ⚠️ **Caution**
>
> User, computer and applications, and character encoding settings may cause an issue when copy-pasting commands or any content from PDF. PDF reader version also affects the copy-pasting functionality. It is recommended to verify the pasted content especially when the hyphens or any special characters are part of the copied content.

## 1.1 Overview

Oracle Communications Operations Services Overlay is a key component of the 5G Service Based Architecture (SBA).

Oracle Communications Operations Services Overlay installs and configures common operation services. For example, you can install and configure Prometheus and its components like AlertManager in a previously installed Kubernetes cluster.

Oracle Communications Operations Services Overlay is an independent deliverable distinct from Oracle Communications Cloud Native Core, Cloud Native Environment (CNE).

## 1.2 References

Following are the reference documents while deploying OSO:

• *Oracle Communications Operations Services Overlay User Guide*

• *Oracle Communications Operations Services Overlay Network Impact Report*

# 2

# Installing OSO

This chapter provides information about installing Oracle Communications Operations Services Overlay (OSO) in a cloud native environment.

> ⓘ **Note**
>
> For release version 24.1.0, OSO supports both fresh installation and upgrade from 23.4.0 and 23.3.x. For more information on how to upgrade OSO, see Upgrading OSO.

OSO, packaged as a Cloud Service Archive (CSAR) file, is a ZIP file that contains the following components:

- All the required OSO images, including opensource software as a tar file.

- All the required OSO Helm charts.

- A custom values.yaml file named `ocoso_24_1_0_0_0-custom-values.yaml`.

> ⓘ **Note**
>
> The README doc contains details to populate mandatory values in the `ocoso_24_1_0_0_0-custom-values.yaml` file. For more information about the configuration parameters, see OSO Configuration Parameters.

## 2.1 Prerequisites

Before installing and configuring OSO, ensure that the following prerequisites are met:

1. CSAR package is downloaded.

2. Unzip and TAR utilities are installed.

3. Docker or Podman is installed, and you must be able to run the docker or podman commands.

4. Helm3 is installed.

5. kubectl is installed.

6. A central repository is made available for all images, binaries, helm charts, and so on, before running this procedure.

7. The following images are populated in the registry on the repo server:

   - `occne.io/oso/prometheus:v2.50.1`

   - `occne.io/oso/alertmanager:v0.26.0`

   - `occne.io/oso/configmapreload:v0.12.0`

   - `occne.io/occne/24_1_common_pod:latest`

All the above images are packed into tar format and will be present in OSO CSAR under (Artifacts/Images) folder. You can use the following commands to load these images into your cluster's registry.

```
$ docker/podman load -i <image-name>.tar
$ docker/podman tag <image-url> <registryaddress>:<port>/image-url
$ docker/podman push <registryaddress>:<port>/image-url
```

# 2.2 Installing OSO Using CSAR

Open Network Automation Platform (ONAP) compliant orchestrator uses CSAR format to onboard, validate, and install OSO. However, in the absence of an orchestrator, manual installation is possible using the CSAR file contents.

Perform the following steps to install OSO using CSAR artifacts:

1. Download the OSO CSAR zip file from My Oracle Support (MOS).

2. Extract the CSAR zip file:

```
$ unzip <OSO CSAR package>
```

3. Upload all the artifacts present in the `Artifacts/Images` folder to the configured repository. For more information about the artifacts, see the Prerequisites section.

4. Update the Prometheus and Alertmanager `ocoso_csar_24_1_0_0_0_custom_values.yaml` file inside `Artifacts/Scripts` directory with the required values as mentioned in OSO Configuration Parameters.

5. Enable IPv6 Dualstack.

   a. Enable IPv6 Dualstack in the `ocoso_csar_24_1_0_0_0_custom_values.yaml`. Search for the following comment in the yaml file and uncomment the following four lines after this comment.

   ```
   # Custom section to enable IPV6, Uncomment below section in order to
   enable Dualstack OSO
        having both Ipv4 and Ipv6 addresses
   #ipFamilies:
   #- IPv4
   #- IPv6
   #ipFamilyPolicy: PreferDualStack
   ```

   Save the file and proceed with the normal installation.

   b. Change the service type from ClusterIP to LoadBalancer to assign IPv6.

      i. Extract the Helm charts tgz file available in the `Artifacts/Scripts` directory.

      ```
      $ cd Artifacts/Scripts
      $ tar -xvzf ocoso_csar_24_1_0_0_0_chart.tgz
      ```

      ii. Install Prometheus and AlertManager using Helm charts provided and updated `ocoso_csar_24_1_0_0_0_custom_values.yaml` file with the following command.

Use the following command for installation using Helm, if custom labels are given:

```
$ kubectl create namespace <deployment-namespace-name>
$ helm install -f <ocoso_csar_24_1_0_0_0_custom_values.yaml> --
namespace=<deployment-namespace-name> --name-template=<deployment-
name> ./prometheus --disable-openapi-validation
```

For Example:

```
$ kubectl create namespace ns1
$ helm install -f <ocoso_csar_24_1_0_0_0_custom_values.yaml> --
namespace=ns1 --name-template=ns1 ./prometheus --disable-openapi-
validation
```

> ⓘ **Note**
>
> Skip the flag (`--disable-openapi-validation`) if custom labels aren't given.

**iii.** Run the following command to perform a Helm test. Populate the values in the `ocoso_csar__custom_values.yaml` file to run the Helm test.

```
$ helm test <release-name> -n <namespace>
```

> ⓘ **Note**
>
> Helm Test can be run for the first time smoothly, but if some issue occurs, and there's a need to re-run the helm test, first you will have to delete the existing test job and repeat the "helm test" command as shown above.
>
> ```
> $ kubectl get jobs.batch -n <namespace>
> ```
>
> ```
> $ kubectl delete jobs.batch oso-test -n <namespace>
> ```

**Figure 2-1    Helm Test**



# 2.3 Postinstallation Tasks

This section explains the postinstallation tasks for OSO.

## 2.3.1 Verifying Installation

To verify if OSO is installed:

1.  Run the following command to verify the OSO version.

    ```
    helm ls -n <oso-namespace>
    ```

    For example:

    ```
    helm ls -n oso-name
    ```

    Sample output:

    ```
    NAME       NAMESPACE        REVISION        UPDATED
    STATUS          CHART                   APP VERSION
    oso     oso          3                   2024-03-27 18:06:42.575679972 +0000 UTC
    deployed        prometheus-15.16.1      24.1.0
    ```

2.  Run the following commands to verify if pods are up and running:

    ```
    $ kubectl get pods --namespace <deployment-namespace-name>
    ```

    For example:

    ```
    $ kubectl get pods -n occne-infra
    ```

Sample output:

```
NAME                          READY   STATUS    RESTARTS   AGE
oso-prom-alm-0                2/2     Running   0          14h
oso-prom-alm-1                2/2     Running   0          14h
oso-prom-svr-84c8c7d488-qsnvx 2/2     Running   0          14h
```

3. Run the following commands to verify if services are up and running and are assigned an EXTERNAL-IP (if LoadBalancer is being used):

```
$ kubectl get service --namespace <deployment-namespace-name>
```

For example:

```
$ kubectl get service -n occne-infra
```

Sample output:

```
NAME                                TYPE          CLUSTER-IP
EXTERNAL-IP     PORT(S)            AGE
oso-prom-alm                        ClusterIP     10.233.16.83
<none>          80/TCP       14h
oso-prom-alm-headless               ClusterIP     None
<none>          80/TCP,6783/TCP    14h
oso-prom-svr                        ClusterIP     10.233.46.136
<none>          80/TCP       14h
```

4. Verify that all the GUIs are accessible.

> ⓘ **Note**
>
> Prometheus and Alertmanager GUIs can be accessed only using the CNC Console. For more information about accessing Prometheus and Alertmanager GUIs using CNC Console, see *Oracle Communications Cloud Native Configuration Console User Guide*.

If the service is of type LoadBalancer, use EXTERNAL-IP to open the Prometheus GUI. Refer to Step 2 to get the services and their EXTERNAL-IPs.

Example to access service IP address with output:

```
# kubectl get nodes -o wide
NAME     STATUS   ROLES    AGE    VERSION   INTERNAL-IP      EXTERNAL-IP
OS-IMAGE                   KERNEL-VERSION                    CONTAINER-RUNTIME
master   Ready    master   87d    v1.17.1   10.75.226.13
&lt;none&gt;          Oracle Linux Server 7.5
4.1.12-112.16.4.el7uek.x86_64   docker://19.3.11
slave1   Ready    &lt;none&gt;   87d    v1.17.1   10.75.225.177
&lt;none&gt;          Oracle Linux Server 7.5
4.1.12-112.16.4.el7uek.x86_64   docker://19.3.11
slave2   Ready    &lt;none&gt;   87d    v1.17.1   10.75.225.47
```

```
&lt;none&gt;        Oracle Linux Server 7.5
4.1.12-112.16.4.el7uek.x86_64   docker://19.3.11
```

```
# kubectl get service -n ocnrf
NAME                                  TYPE          CLUSTER-IP
EXTERNAL-IP     PORT(S)          AGE
oso-prom-alm                          ClusterIP     10.103.63.10
&lt;none&gt;         80/TCP           35m
oso-prom-alm-headless                 ClusterIP     None
&lt;none&gt;         80/TCP           35m
oso-prom-svr                          ClusterIP     10.101.91.81
&lt;none&gt;         80/TCP           35m
```

> ⓘ **Note**
>
> In case Dualstack with IPv6 is enabled, change the service type to LoadBalancer
> in the custom values file. You can see both IPs when calling the `kubectl get`
> `service` command, and the output is as follows:
>
> ```
> [root@master Scripts]# kubectl get service -n ocnrf
> NAME                          TYPE           CLUSTER-IP
> EXTERNAL-IP                          PORT(S)          AGE
> oso-prom-alm                  LoadBalancer   10.103.63.10
> 10.75.202.205,2606:b400:605:b809::2  80/TCP           35m
> oso-prom-alm-headless         ClusterIP      None
> <none>                               80/TCP           35m
> oso-prom-svr                  LoadBalancer   10.101.91.81
> 10.75.202.204,2606:b400:605:b809::1  80/TCP           35m
> ```

**Figure 2-2    Prometheus GUI**
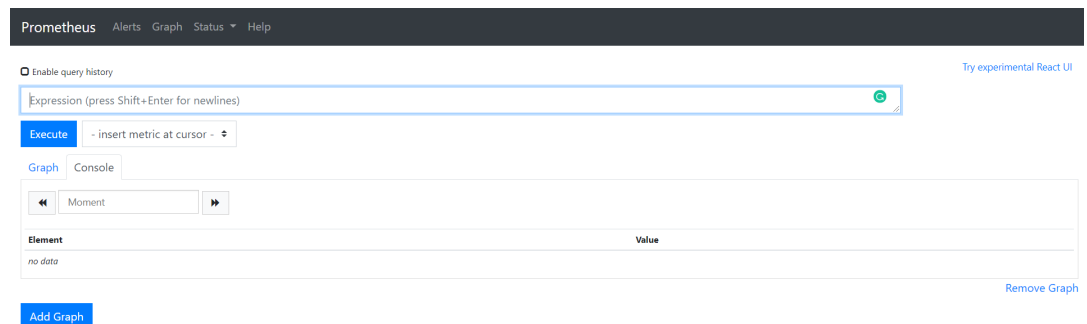
**Figure 2-3    Alert Manager GUI**
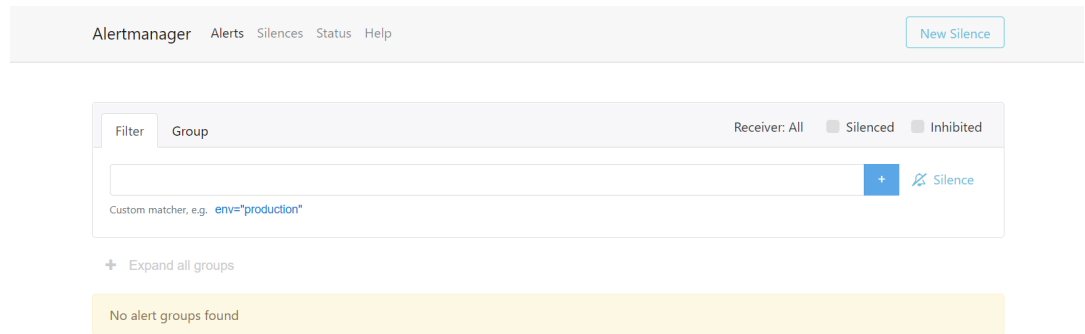


5. On the Prometheus GUI, click **Alerts** to verify that all the alerts (NF Alerts) are visible.

> ⓘ **Note**
>
> OSO Prometheus doesn't have any alerts of its own, therefore the GUI must appear empty initially. You can patch any NF alert rules in this section.

The following image displays the alerts for cnDBTier:

**Figure 2-4    Prometheus GUI**
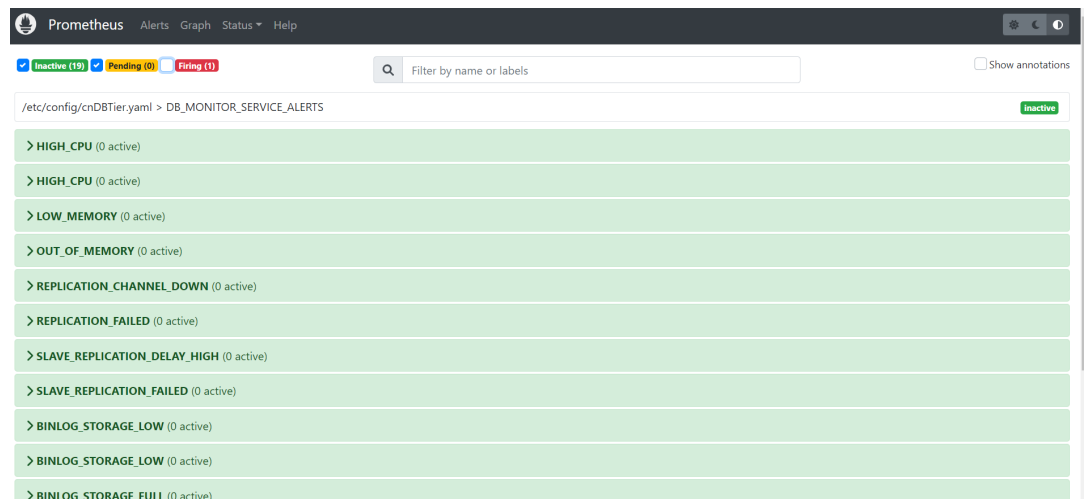


After alerts are raised, the GUI must display the triggered alerts as shown in the following image:

Figure 2-5    Prometheus GUI - Alerts
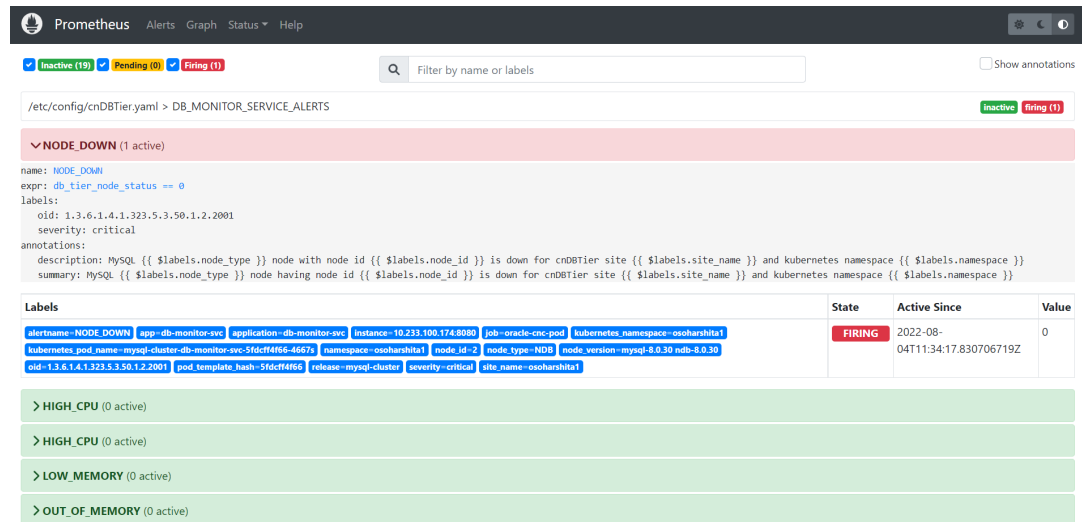


6.  Select **Alerts** tab on the Alertmanager GUI to view the triggered alerts as shown in the following image:

Figure 2-6    Alertmanager - Alerts
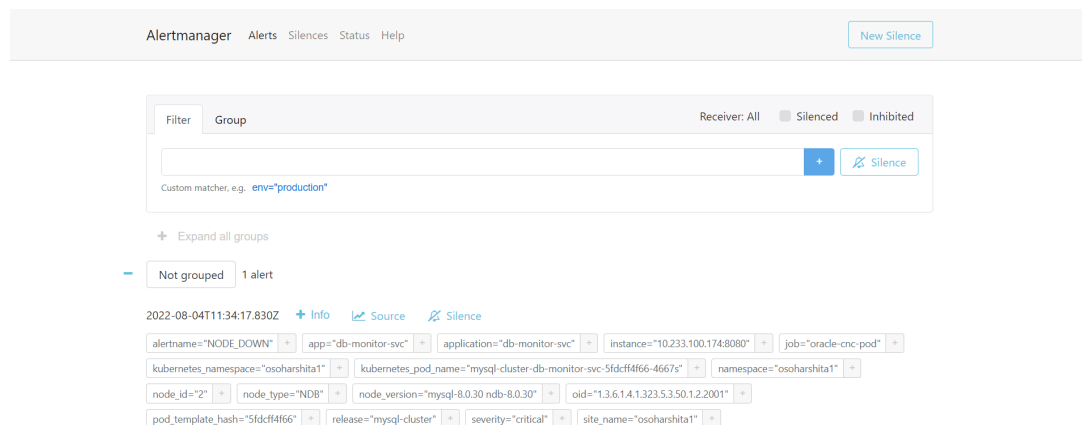


7.  On the Prometheus GUI, click **Graph** to verify if the expected metrics (example, NF Metrics) are displayed. The following image displays a sample Prometheus Graph with metrics:

**Figure 2-7    Prometheus Graph**
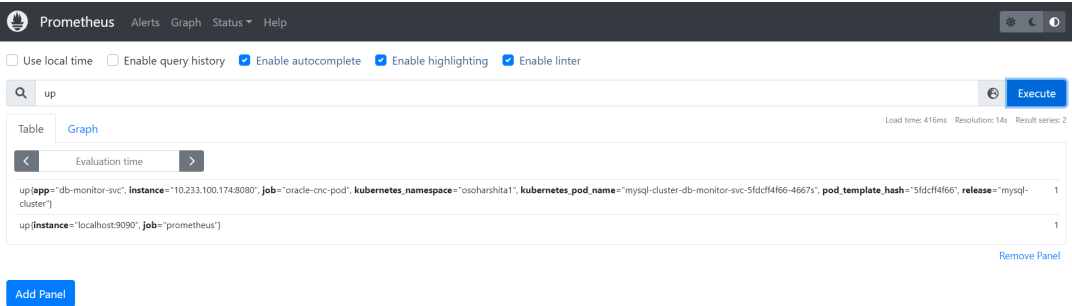


8. On the Prometheus GUI, navigate to **Status** and then click **Target** to verify if the configured targets are displayed.
   The following image shows Prometheus targets that are being extracted:

**Figure 2-8    Prometheus Target**
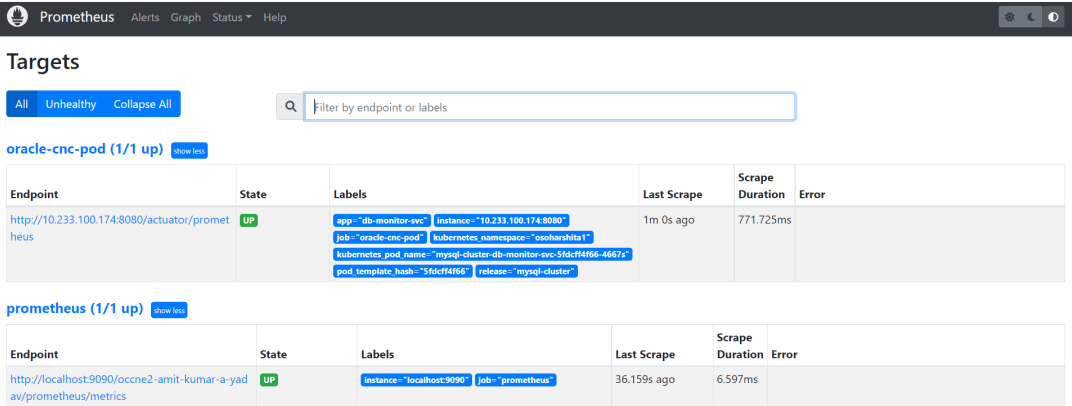
# 3

# Customizing OSO

This chapter provides information about customizing Oracle Communications Operations Services Overlay (OSO) deployment in a cloud native environment.

The OSO deployment is customized by overriding the default values of various configurable parameters in the `<custom-value.yaml>` file.

## 3.1 Naming Conventions and Limits

This section provides the naming conventions and limits for OSO containers and labels.

**Naming Conventions and Limits for Containers**

OSO supports customization of container names using custom prefix and suffix. Currently, the name field supports up to 63 characters (including the custom prefix, custom suffix, and default fixed name).

> ⓘ **Note**
>
> - The maximum length of default fixed names is 17.
>
> - To avoid name truncation, limit the number of characters in the custom prefix and suffix to match the defined limit.

The following table displays the default fixed names of containers without prefix and suffix:

**Table 3-1    Default Fixed Names of Containers**

| POD Name | Fixed Container Name |
|---|---|
| Prometheus-server(prom-svr) | • prom-svr-cm-r<br>• prom-svr |
| Alertmanager(prom-alm) | • prom-alm<br>• prom-alm-cm-r |

You can use the following values in the `oso custom values.yaml` file to add the custom prefix and suffix to container names:

```
global:
# provide the prefix and suffix to be added to k8Resources
  k8Resource:
    container:
      prefix:
      suffix:
```

**Naming Conventions and Limits for Labels**

1. OSO supports custom labels for Kubernetes resources. It provides three types of label definitions that must be configured in the `oso custom values.yaml` file:

   a. **Global Labels(allResources)**: These labels are attached to all the Kubernetes resources. This is depicted as TYPE1 in the following sample `oso custom values.yaml` file.

   b. **LB and NON-LB TYPE label**: These labels are attached to LoadBalancer and non-LoadBalancer type deployments, statefulsets, and services. This is depicted as TYPE2 in the following sample `oso custom values.yaml` file.

   c. **Service specific label**: These labels are attached to each service in their specific service-label metadata sections and can be used to uniquely label services, regardless of them being LoadBalancer or non-LoadBalancer.

   > ⓘ **Note**
   >
   > Prometheus and Alertmanager have their own set of service-specific labels.

   You can use the following values in the `oso custom values.yaml` file to provide custom labels for Kubernetes resources:

   ```
   customExtension:
   # TYPE1 Label
       allResources:
           labels: {}
   # TYPE2 Labels
       lbServices:
           labels: {}

       nonlbServices:
           labels: {}

       lbDeployments:
           labels: {}

       nonlbDeployments:
           labels: {}

       lbStatefulSets:
           labels: {}
   ..
   ```

   > ⓘ **Note**
   >
   > You must be well aware of the naming guidelines while labeling the resources. For more information on Helm naming guidelines, see [Helm Labels and Annotations](Helm Labels and Annotations).

# 3.2 OSO Configuration Parameters

**Overview**

This section provides the configuration parameters details that are used to install the OSO CSAR packages manually.

OSO CSAR is packaged with the following open source images:

- prometheus
- alertmanager
- configmapreload
- 24_1_common_pod

**Prerequisites**

- Docker or Podman must be installed.
- Helm repository must be set up and accessible.

**Table 3-2    OSO Configuration Parameter**

| Parameter | Description | Details |
|-----------|-------------|---------|
| OSO-RELEASE-NAME | This is a mandatory parameter.<br>Indicates the Helm release name for the OSO release.<br>Example: promsvcname: oso-prom-svralmsvcname: oso-prom-alm | **Data Type:** String<br>**Default Value:** NA<br>**Range:** NA |
| USEASM | This is a conditional parameter.<br>Indicates if the Aspen Service Mesh (ASM) environment is used or not.<br>Example: useasm: true | **Data Type:** Boolean<br>**Default Value:** false<br>**Range:** true, false |
| NAMESPACE | This is a mandatory parameter.<br>Indicates the namespace where OSO will be installed. This value is required in multiple places in the custom values.yaml file.<br>Example: namespace: ocnf | **Data Type:** String<br>**Default Value:** NA<br>**Range:** NA |
| CLUSTER-NAME | This is a mandatory parameter.<br>Indicates the name of the cluster where OSO is deployed. This value is required in multiple places in the custom values.yaml file.<br>Example: clustername: oso-cluster | **Data Type:** String<br>**Default Value:** NA<br>**Range:** NA |

**Table 3-2    (Cont.) OSO Configuration Parameter**

| Parameter | Description | Details |
|---|---|---|
| `serviceAccountNamePromSvr` | This is a mandatory parameter.<br><br>Indicates the ServiceAccount that is to be used by the Prometheus resource.<br><br>Users must create ServiceAccount with the required namespace access defined along with the Role or RoleBinding for the Prometheus service.<br><br>Example:<br><br>`# Service account for Prometheus`<br>`  serviceAccountNamePromSvr: "oso-prom"`<br><br>If you do not have a ServiceAccount, then set the `USE-DEFAULT-SVCACCOUNT-PROM` parameter to true. | **Data Type:** String<br>**Default Value:**""<br>**Range:** NA |
| `serviceAccountNameAlertMgr` | This is a mandatory parameter.<br><br>Indicates the ServiceAccount that is to be used by the Alertmanager resource.<br><br>Users must create ServiceAccount with the required namespace access defined along with the Role or RoleBinding for the Alertmanager service.<br><br>Example:<br><br>`# Service account for Alertmanagers`<br>`  serviceAccountNameAlertMgr: "oso-alm"`<br><br>If you do not have a ServiceAccount, then set the `USE-DEFAULT-SVCACCOUNT-PROM` parameter to true. | **Data Type:** String<br>**Default Value:**""<br>**Range:** NA |
| `ALERTMANAGER-REPO-ADDRESS` | This is a mandatory parameter.<br><br>Indicates the repo URL for alertmanager location.<br><br>Example: `Bastion:5000/occne.io/oso/alertmanager` | **Data Type:** String<br>**Default Value:** NA<br>**Range:** NA |
| `PROMETHEUS-REPO-ADDRESS` | This is a mandatory parameter.<br><br>Indicates the docker image repo URL for Prometheus location.<br><br>Example: Bastion:5000/occne.io/oso/prometheus | **Data Type:** String<br>**Default Value:** NA<br>**Range:** NA |

**Table 3-2    (Cont.) OSO Configuration Parameter**

| Parameter | Description | Details |
|---|---|---|
| `HELM-TEST-IMAGE-REPO-ADDRESS` | This is a mandatory parameter.<br>Indicates the repo URL of the Helm test image location.<br>Example:<br><br>`Bastion:5000/occne.io/occne/`<br>`24_1_common_pod` | **Data Type:** String<br>**Default Value:** NA<br>**Range:** NA |
| `RETENTION_TIME_ALM` | This is a mandatory parameter.<br>Indicates the amount of time to retain the alertmanager data.<br>Example:<br><br>`extraArgs:`<br>`    data.retention: 120h` | **Data Type:** String<br>**Default Value:** NA<br>**Range:** NA |
| `ALM-PV-SIZE` | This is a mandatory parameter.<br>Indicates the size of the persistence volume of alertmanager.<br>Example:<br><br>`persistentVolume:`<br>`    enabled: true`<br>`    ...`<br>`    size: 2Gi` | **Data Type:** String<br>**Default Value:** NA<br>**Range:** NA |
| `ALM-STORAGE-CLASS` | This is a mandatory parameter.<br>Indicates the storage class for alertmanager.<br><br>`persistentVolume:`<br>`    enabled: true`<br>`    .......`<br>`    storageClass: standard` | **Data Type:** String<br>**Default Value:**standard<br>**Range:**NA |
| `CONFIGMAP-RELOAD-REPO-ADDRESS` | This is a mandatory parameter.<br>Indicates the docker image repo URL for configmap-reload location.<br>Example:<br><br>`Bastion:5000/occne.io/oso/`<br>`configmapreload` | **Data Type:** String<br>**Default Value:** NA<br>**Range:** NA |

**Table 3-2    (Cont.) OSO Configuration Parameter**

| Parameter | Description | Details |
|---|---|---|
| `server:` `.......` `  ##` `namespace` `s to` `monitor(i` `nstead` `of` `monitorin` `g all -` `clusterwi` `de).` `##Needed` `if you` `want to` `run` `without` `Cluster-` `admin` `privilege` `s.` `namespace` `s:` `  -` `NAMESPACE` | This is a mandatory parameter.<br>If clusterrole is used, comment the namespaces section. If role is used, then give the namespace where OSO is deployed.<br>Example:<br><br>`server:`<br>`  .......`<br>`   ## namespaces to`<br>`monitor(instead of monitoring`<br>`all - clusterwide).`<br>`   ## Needed if you want to run`<br>`without Cluster-admin`<br>`privileges.`<br>`   ## namespaces:`<br>`   ## - NAMESPACE` | **Data Type:** String<br>**Default Value:** NA<br>**Range:** NA |
| `BEARER_TOK` `EN` | This is a conditional parameter.<br>Set the Authorization header on every remote write request with the configured bearer token. In this case, it is Cortex Server.<br>Example:<br><br>`remote_write`<br>`# bearer_token for cortex`<br>`server to be configured`<br>`bearer_token:`<br>`eyJhbGciOiJIUzUxMiIsInR5cCI6Ikp`<br>`XVCJ9.` | **Data Type:** String<br>**Default Value:** NA<br>**Range:** NA |

**Table 3-2    (Cont.) OSO Configuration Parameter**

| Parameter | Description | Details |
|---|---|---|
| STORAGE_RE TENTION_SI ZE_PROM | This is a mandatory parameter.<br><br>Indicates the retention size for Prometheus persistence volume.<br>Example:<br><br>`extraArgs:`<br><br>`storage.tsdb.retention.size: 2GB` | **Data Type:** String<br>**Default Value:** NA<br>**Range:** NA |
| PROM-PV-SIZE | This is a mandatory parameter.<br>Indicates the size for the persistence volume of Prometheus.<br>Example:<br><br>`persistentVolume:`<br>`    enabled: true`<br>`    size: 8Gi` | **Data Type:** String<br>**Default Value:** NA<br>**Range:** NA |
| PROM-STORAGE-CLASS | This is a mandatory parameter.<br>Indicates the storage class for Prometheus.<br>Example:<br><br>` persistentVolume:`<br>`    enabled: true`<br>`    ......`<br>`    storageClass: standard` | **Data Type:** String<br>**Default Value:**standard<br>**Range:** NA |
| PROM-DATA-RETENTION-DAYS | This is a mandatory parameter.<br>Indicates the retention days for prometheus persistence volume.<br>Example:<br><br>`retention: 7d` | **Data Type:** String<br>**Default Value:** NA<br>**Range:** NA |
| USE-DEFAULT-SVCACCOUNT-ALM | This is a mandatory parameter.<br>If default svcaccount is required at cluster scope, set this parameter to true, else set it to false and create your own svcaccount and configure OSO.<br>Example:<br><br>`serviceAccounts:`<br>`  alertmanager:`<br>`    create: false` | **Data Type:** Boolean<br>**Default Value:** false<br>**Range:** true, false |

**Table 3-2 (Cont.) OSO Configuration Parameter**

| Parameter | Description | Details |
|---|---|---|
| USE-DEFAULT-SVCACCOUNT-PROM | This is a mandatory parameter.<br><br>If default svcaccount is required at cluster scope, then set this parameter to true, else set it to false and create your own svcaccount and configure OSO.<br>Example:<br><br>```serviceAccounts:<br>  prometheus:<br>    create: false``` | **Data Type:** Boolean<br>**Default Value:** false<br>**Range:** true, false |
| OSO_CORTEX_URL | This is a conditional parameter.<br>Indicates the URL where Prometheus writes metrics data.<br>Example:<br><br>```http://<some-ip>:<some-port>/api/prom/push``` | **Data Type:** String<br>**Default Value:** NA<br>**Range:** NA |
| OSO_REMOTE_WRITE_TIMEOUT | This is a conditional parameter.<br>Indicates the maximum amount of time Prometheus waits for a response to the remote write request.<br>Example: 30s | **Data Type:** String<br>**Default Value:** NA<br>**Range:** NA |
| namespaces | This is a conditional parameter.<br><br>If you are using external Service Account with namespace specific Role or RoleBinding Access, the namespaces must be configured in Prometheus for extracting the given allowed namespaces.<br><br>Uncomment the following lines in the custom values.yaml file and replace ns1 and ns2 with namespaces you want to configure for extracting, in each of the Prometheus jobs.<br><br>```- job_name: 'oracle-cnc-pod'<br>kubernetes_sd_configs:<br>- role: pod<br>#namespaces:<br># names:<br># - ns1<br># - ns2```<br><br>You can add more namespaces in the list as per your requirement. | **Data Type:** String<br>**Default Value:** NA<br>**Range:** NA |

**Table 3-2    (Cont.) OSO Configuration Parameter**

| Parameter | Description | Details |
|---|---|---|
| Prefix Suffix | This is a conditional parameter.<br>Indicates the prefix and suffix that will be added to K8s Resource names.<br>Example:<br><br>```<br>global:<br>  k8Resource:<br>    container:<br>      prefix: oracle1<br>      suffix: oracle2<br>```<br><br>**Note:** Currently, only Containers are configured with prefix and suffix. | **Data Type:** String<br>**Default Value:** NA<br>**Range:** NA |
| `allResources: labels: {}` | This is a conditional parameter.<br>Indicates the Global Labels (allResources) that will be attached to all Kubernetes resources.<br>Example:<br><br>```<br>allResources:<br>     labels:<br><br>          env : production<br>```<br> | **Data Type:** String<br>**Default Value:** NA<br>**Range:** NA |

**Table 3-2    (Cont.) OSO Configuration Parameter**

| Parameter | Description | Details |
|---|---|---|
| `lbServices:`<br>` labels:`<br>`{}`<br>`nonlbServices:`<br>` labels:`<br>`{}`<br>`lbDeployments:`<br>` labels:`<br>`{}`<br>`nonlbDeployments:`<br>` labels:`<br>`{}`<br>`lbStatefulSets:`<br>` labels:`<br>`{}` | This is a conditional parameter.<br>Indicates the LB and non-LB TYPE labels that will be attached to the load balancer and non-load balancer type deployments, statefulsets, and services.<br>Example:<br><br>`lbServices:`<br>`        labels:`<br><br>`            key1 : value1`<br><br>`    nonlbServices:`<br>`      labels:`<br>`          key2 : value2`<br><br>`    lbDeployments:`<br>`      labels:`<br>`          key3 : value3`<br><br>`    nonlbDeployments:`<br>`      labels:`<br>`          key4 : value4`<br><br>`    lbStatefulSets:`<br>`      labels:`<br>`          key5 : value5` | **Data Type:** String<br>**Default Value:** NA<br>**Range:** NA |
| `annotations: {}` | This is a conditional parameter.<br>Specifies the custom annotations to be added to OSO deployments that are associated to disable the sidecar injections into OSO pods.<br>Example:<br><br>`annotations:`<br>`    - sidecar.istio.io/inject:`<br>`"false"` | **Data Type:** Boolean<br>**Default Value:** false<br>**Range:** true, false |

**Table 3-2    (Cont.) OSO Configuration Parameter**

| Parameter | Description | Details |
|---|---|---|
| ```
alertmana
gers:
  -
kubernete
s_sd_conf
igs:
    -
role: pod
   #
namespace
 to be
configure
d

namespace
s:

 names:
{}

relabel_c
onfigs:
   -
source_la
bels:
[__meta_k
ubernetes
_namespac
e]
   #
namespace
 to be
configure
d

regex: {}
``` | This is a mandatory parameter.<br>Indicates if this parameter is used to add namespaces for Alertmanager.<br>Example:<br><br>```
alertmanagers:
  - kubernetes_sd_configs:
      - role: pod
# namespace to be configured
        namespaces:
          names:
          - ns1
    relabel_configs:
    - source_labels:
[__meta_kubernetes_namespace]
       regex: ns1
``` | **Data Type:** String<br>**Default Value:** NA<br>**Range:** NA |

# 4
# Using OSO with IPv6

This section details the steps to access OSO services like Prometheus and Alertmanager using IPv6.

> ⓘ **Note**
>
> You can use the Installing OSO using CSAR procedure to install OSO in an IPv6 cluster.

- To access Prometheus GUI using IPv6, use the following URL format and open the GUI using any IPv6 supported browser:

  Load Balancer:

  ```
  LoadBalancerIP-Prometheus]
  ```

  Example:

  ```
  http://[2606:b400:605:b819:4631:92ff:fe73:9d2c]
  ```

  Node Port:

  ```
  [AnyWorkerNodeIP]:NodePort-Prometheus
  ```

  Example:

  ```
  http://[2606:b400:605:b819:4631:92ff:fe73:9d2c]:30090
  ```

- Currently, Alertmanager has an issue in accessing Alertmanager GUI with IPv6 (see, Parsing URLs with IPv6 host issue). To overcome this issue, perform the following steps to use the DNS names concept as Alertmanager does not support IPv6 address format.

  1. Open the `hosts` file from the following path:
     `C:\Windows\System32\drivers\etc\hosts`.

  2. Add the following entry to map the worker-node IP of IPv6 cluster with a DNS name:

     ```
     # DNS Mapping for alertmanager
     2606:b400:605:b819:4631:92ff:fe73:9d2c alertmanager
     ```

  3. Open GUI using NodePort and DNS name. For example, `http://alertmanager:30854/alertmanager`

# 5

# Upgrading OSO

This chapter provides information about upgrading Oracle Communications Operations Services Overlay (OSO) deployment to the latest release. It is recommended to perform OSO upgrade in a specific order. For more information about the upgrade order, see *Oracle Communications Cloud Native Core, Solution Upgrade Guide*.

> ⓘ **Note**
>
> While performing OSO upgrade from 23.3.x and 23.4.0 to 24.1.0, a small outage is expected.

## 5.1 Supported Upgrade Paths

The following table lists the supported upgrade paths for OSO:

**Table 5-1    Supported Upgrade Path**

| Source Release | Target Release |
|---|---|
| 23.4.0 | 24.1.0 |
| 23.3.1 | 24.1.0 |

## 5.2 Prerequisites

Following are the prerequisites for upgrading OSO:

- Ensure that the system has OSO 23.4.0 or 23.3.x installed and the OSO is working properly.

- Verify if all the pods and services are up and running.

- Ensure that the NFs are upgraded after the OSO upgrade.

## 5.3 Upgrading OSO using CSAR

This section describes the procedure to upgrade an existing OSO using CSAR.

1.  Prepare the `ocoso_csar_24_1_0_0_0_custom_values.yaml` file by performing the following steps. These .yaml file is used as the input during the upgrade process.

    a.  Download OSO CSAR zip file from [My Oracle Support](#) (MOS).

    b.  Unzip the CSAR zip file:

        ```
        $ unzip <OSO CSAR package>
        ```

   **c.** Upload all the artifacts present in the `Artifacts/Images` folder to the configured repository. For more information on the artifacts, see the [Installation Prerequisite](#) section.

   **d.** Update the Prometheus and Alertmanager `ocoso_csar_24_1_0_0_0_custom_values.yaml` file present in the `Artifacts/Scripts` directory with the required values as mentioned in [OSO Configuration Parameters](#).

   **e.** Untar the Helm charts tgz file available in the `Artifacts/Scripts` directory.

```
cd Artifacts/Scripts
$ tar -xvzf <helm charts.tgz file>
```

2. Run the following command to upgrade the existing OSO release:

```
$ helm upgrade <release-name> <path to the OSO helm chart> -n <namespace> -
f <path to ocoso_csar_24.1.0_custom_values.yaml file>
```

3. Post the Helm upgrade a new Prometheus pod spuns up, which is in container creating state. To make it into running state, detach the PVC from old pod in K8s and let it attach to this new upgraded pod.

```
$ kubectl get deployment -n <oso-namespace>
# Find Prometheus deployment name and replace it in below commands, Both
the commands below needs to be run at the same time
$ kubectl -n <oso-namespace> scale deploy <oso-deployment-name> --
replicas=0
$ kubectl -n <oso-namespace> scale deploy <oso-deployment-name> --
replicas=1
```

4. Verify if all the pods are up and running with the latest versions and changes.

5. Verify if the upgraded OSO is working as expected by performing the [Verifying Installation](#) procedure.

# 6

# Rolling Back OSO

This chapter provides information about rolling back Oracle Communications Operations Services Overlay (OSO) deployment to previous releases.

## 6.1 Supported Rollback Paths

The following table lists the supported rollback paths for OSO:

**Table 6-1    Supported Rollback Path**

| Source Release | Target Release |
|---|---|
| 24.1.0 | 23.4.0 |
| 24.1.0 | 23.3.1 |

## 6.2 Prerequisites

Following are the prerequisites to rollback OSO:

- Ensure that the system has OSO 24.1.0 installed and the OSO is working properly.

- Verify if all the pods and services are up and running.

## 6.3 Rollback Tasks

To roll back from OSO 24.1.0 to previous releases 23.3.x or 23.4.0:

1. Check the revision you want to roll back your release to.

    ```
    $ helm -n <namespace> history <oso-release-name>
    ```

    For example:

    ```
    $ helm -n dbtier history oso
    ```
    Sample output

    ```
    REVISION        UPDATED                         STATUS
    CHART                   APP VERSION             DESCRIPTION
    1               Mon Mar 25 17:46:51 2024        superseded
    prometheus-15.16.1      24.1.0-62-g28668fe Install complete
    2               Mon Mar 25 17:48:01 2024        deployed
    prometheus-15.16.1      24.1.0-62-g28668fe Upgrade complete
    ```

2. Pick the Revision number to which you wish to roll back your release into. In above example, it will be 1. Run the following command to roll back:

```
$ helm  -n <namespace> rollback <oso-release-name> <oso revision number>
```

For example:

```
$ helm -n dbtier rollback oso 1
```
Sample output

```
Rollback was a success! Happy Helming!
```

3. Verify if the rollback was successful to a previous version, perform the following command:

```
$ helm -n <namespace> history <oso-release-name>
```

For example:

```
$ helm -n dbtier history oso
```
Sample output

```
REVISION        UPDATED                         STATUS
CHART                   APP VERSION             DESCRIPTION
1               Mon Mar 25 17:46:51 2024        superseded
prometheus-15.16.1      24.1.0-62-g28668fe Install complete
2               Mon Mar 25 17:48:01 2024        superseded
prometheus-15.16.1      24.1.0-62-g28668fe Upgrade complete
3               Wed Mar 27 18:37:02 2024        deployed
prometheus-15.16.1      24.1.0-62-g28668fe Rollback to 1
```

# A

# Key Information About OSO

This section provides important information that the user must know about OSO.

The following are some of the important points about OSO installation that you must be aware of:

1. All the services having GUI are converted to ClusterIP as Cloud Native Configuration Console (CNC Console) hyperlinks for OSO GUIs are currently not enabled. Therefore, you can access the GUIs through CNC Console only.
   Run the following command to get the list of services and check their service type.

   ```
   $ kubectl get service -n <namespace>
   ```

   Sample output:

   ```
   NAME                          TYPE         CLUSTER-IP       EXTERNAL-
   IP        PORT(S)          AGE
   oso-prom-alm                  ClusterIP    10.233.48.28
   <none>        80:31717/TCP       4d13h
   oso-prom-alm-headless         ClusterIP    None
   <none>        80/TCP,6783/TCP    4d13h
   oso-prom-svr                  ClusterIP    10.233.22.117
   <none>        80:30266/TCP       4d13h
   ```

2. Since Alertmanager is a StatefulSets, the Persistent Volume (PV) or Persistent Volume Claim (PVC) do not get deleted automatically when the following helm uninstallation command is run for the respective OSO helm-release:

   ```
   $ helm3 uninstall oso-release-name -n <namespace>
   ```

   This uninstallation command removes all resources related to the OSO release. However it does not remove the PVs or PVCs for Alertmanager. Therefore, perform the following steps to manually remove PVs and PVCs in Alertmanager:

   a. Run the following command to retrieve the list of PVCs:

   ```
   $ kubectl get pvc -n <namespace>
   ```

   Sample output:

   ```
   NAME             STATUS    VOLUME
   CAPACITY    ACCESS MODES    STORAGECLASS    AGE
   oso-prom-svr     BOUND     pvc-a639c8cb-eee1-4fa3-be1b-d4c4e5d66b9a
   8Gi          RWO          standard        159m
   oso-prom-alm-0   BOUND     pvc-a639c8cb-eee1-4fa3-be1b-d4c4e5d6689J
   2Gi          RWO          standard        159m
   ```

```
oso-prom-alm-1    BOUND      pvc-a639c8cb-eee1-4fa3-be1b-d4c4e5d6689k
2Gi         RWO          standard         159m
```

**b.** Run the following command to delete the PVCs:

```
$ kubectl delete pvc oso-prom-svr oso-prom-alm-0 oso-prom-alm-1 -n
<namespace>
```

Sample output:

```
persistentvolumeclaim "oso-prom-svr" deleted
persistentvolumeclaim "oso-prom-alm-0" deleted
persistentvolumeclaim "oso-prom-alm-1" deleted
```

**c.** Run the following command to retrieve the PV details:

```
$ kubectl get pv -n <namespace>
```

Sample output:

```
NAME                                      CAPACITY   ACCESS MODES
RECLAIM POLICY    STATUS
CLAIM
STORAGECLASS      REASON    AGE
pvc-104a1c0c-72f7-4ae0-b368-71c4e22082ce   2Gi          RWO
Delete            Bound    osoxyz2/storage-volume-oso-xyz2-prom-
alm-1           standard                  4d13h
pvc-1e75d93e-0731-45f2-9ef7-46cf8cbbb3f9   2Gi          RWO
Delete            Bound    osoxyz1/storage-volume-oso-xyz-prom-
alm-0           standard                  4d13h
pvc-30400fe7-61dd-4e2f-8a50-149e4e14be90   2Gi          RWO
Delete            Bound    osoxyz2/oso-xyz2-prom-
svr                        standard                    4d12h
```

**d.** Run the following command to delete the PVs:

```
$ kubectl delete pv pvc-104a1c0c-72f7-4ae0-b368-71c4e22082ce
pvc-1e75d93e-0731-45f2-9ef7-46cf8cbbb3f9
pvc-30400fe7-61dd-4e2f-8a50-149e4e14be90 -n osoxyz1
```

**3.** Alertmanager has persistent storage enabled in this release to store notifications and silences.