Oracle® Communications

Cloud Native Core, Operations Services Overlay Installation and Upgrade Guide

Release 25.1.201
G33179-03
December 2025

ORACLE

Oracle Communications Cloud Native Core, Operations Services Overlay Installation and Upgrade Guide, Release 25.1.201

G33179-03

Contents

Preface

Documentation Accessibility

For information about Oracle's commitment to accessibility, visit the Oracle Accessibility Program website at http://www.oracle.com/pls/topic/lookup?ctx=acc&id=docacc.

Access to Oracle Support

Oracle customer access to and use of Oracle support services will be pursuant to the terms and conditions specified in their Oracle order for the applicable services.

Diversity and Inclusion

Oracle is fully committed to diversity and inclusion. Oracle respects and values having a diverse workforce that increases thought leadership and innovation. As part of our initiative to build a more inclusive culture that positively impacts our employees, customers, and partners, we are working to remove insensitive terms from our products and documentation. We are also mindful of the necessity to maintain compatibility with our customers' existing technologies and the need to ensure continuity of service as Oracle's offerings and industry standards evolve. Because of these technical constraints, our effort to remove insensitive terms is ongoing and will take time and external cooperation.

Conventions

The following text conventions are used in this document:

Convention	Meaning
boldface	Boldface type indicates graphical user interface elements associated with an action, or terms defined in text or the glossary.
italic	Italic type indicates book titles, emphasis, or placeholder variables for which you supply particular values.
`monospace`	Monospace type indicates commands within a paragraph, URLs, code in examples, text that appears on the screen, or text that you enter.

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
OSDC	Oracle Software Download Center
OSO	Operations Services Overlay
OSO_PROM_CONFIGMAP	Configmap name of OSO prometheus
OSO_ALERTRULE_CONFIG_FILE	Alert-rule file of NFs
OSO_ALR_CONFIG_CONFIGMAP	Configmap name of OSO-ALR-CONFIG feature
prom	Prometheus
svr	Server
TSDB	Time Series Database

What's New in This Guide

This section introduces the documentation updates for release 25.1.2xx.

Release 25.1.201- G33179-03, December 2025

General Updates

Updated the release version to 25.1.201 in the entire document.

Installation Updates

Updated the versions for the following images in the Prerequisites section:

- `OSO_ALERT_CONFIG_IMAGE=occne.io/occne/oso_alert_config:25.1.201`
- `OSO_HELM_TEST_IMAGE=occne.io/occne/oso_helm_test:25.1.201`
- `OSO_SNAPSHOT_IMAGE=occne.io/occne/oso_snapshot:25.1.201`

Upgrade and Rollback Updates

- Updated the upgrade paths in the Supported Upgrade Paths section.
- Updated the current release and the previous release versions in the Prerequisites section.
- Updated that OSO does not support rollback to previous versions in the Rolling Back OSO section.

Release 25.1.200- G33179-02, September 2025

General Updates

Updated the procedures in the Creating Backup of Prometheus Time Series Database (TSDB) Using Snapshot Utility section.

Installation Updates

Updated the resource utilization table with the Ephemeral Storage Request and Ephemeral Storage Limit in the Resource Utilization section.

Release 25.1.200- G33179-01, July 2025

General Updates

- Updated the release version to 25.1.200 in the entire document.
- Updated the release information in the Table 1-1 in the Oracle Error Correction Policy section.

Installation Updates

- Updated the versions for the following images in the Prerequisites section:
  - `OSO_ALERTMANAGER_IMAGE=occne.io/oso/alertmanager:v0.28.0`
  - `OSO_CONFIGMAPRELOAD_IMAGE=occne.io/oso/configmapreload:v0.15.0`
  - `OSO_PROMETHEUS_IMAGE=occne.io/oso/prometheus:v3.2.0`
  - `OSO_SNAPSHOT_IMAGE=occne.io/occne/oso_snapshot:25.1.200`

- Added the Automated Configuration of NF Alerts section with the procedure for configuring NF alerts using automation.
- Added the `ocoso_csar_25_1_200_0_0_0_alr_config_custom_values.yaml` parameters in the Installing OSO section.
- Added the Alert Config Configuration Parameters section with the alert configuration file parameters.
- Added the `rule_files` parameter in the Prometheus Configuration Parameters section.
- Added the following images in the Prerequisites section:
  - `OSO_ALERT_CONFIG_IMAGE =occne.io/occne/oso_alert_config:25.1.200`
  - `OSO_HELM_TEST_IMAGE=occne.io/occne/oso_helm_test:25.1.200`

Upgrade and Rollback Updates

- Updated the upgrade paths in the Supported Upgrade Paths section.
- Updated the current release and the previous release versions in the Prerequisites section.
- Updated that OSO does not support rollback to previous versions in the Rolling Back OSO section.

1
Introduction

This guide describes how to install or upgrade Oracle Communications Cloud Native Core, 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 Cloud Native Core, Operations Services Overlay is a key component of the 5G Service Based Architecture (SBA).

Oracle Communications Cloud Native Core, 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 Cloud Native Core, 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 Cloud Native Core, Operations Services Overlay User Guide*
- *Oracle Communications Cloud Native Core, Operations Services Overlay Network Impact Report*
- *Oracle Communications Cloud Native Core Release Notes*
- *Oracle Communications Cloud Native Core Licensing Information User Guide*
- *Oracle Communications Cloud Native Core Solution Upgrade Guide*
- *Oracle Communications Cloud Native Core Security Guide*

1.3 Oracle Error Correction Policy

The table below outlines the key details for the current and past releases, their General Availability (GA) dates, the latest patch versions, and the end dates for the Error Correction Grace Period.

Table 1-1 Error Correction Policy

Cloud Native Core Release Number	General Availability (GA) Date	Error Correction Grace Period End Date
3.25.1.200	July 2025	July 2026
3.25.1.100	April 2025	April 2026
3.24.3	October 2024	October 2025
3.24.2	July 2024	July 2025

> **Note**
>
> - For the latest patch releases, see their corresponding *Oracle Communications Cloud Native Core Release Notes*.
> - For a release, Sev1 and Critical Patch Unit (CPU) patches are supported for 12 months. For more information, see Oracle Communications Cloud Native Core and Network Analytics Error Correction Policy.

1.4 Oracle Open Source Support Policies

Oracle Communications Cloud Native Core uses open source technology governed by the Oracle Open Source Support Policies. For more information, see Oracle Open Source Support Policies.

2
Installing OSO

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

> **Note**
>
> For release version 25.1.201, OSO supports both fresh installation and upgrade from 25.1.200 and 25.1.1xx. 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.
- The following custom values.yaml files:
  - `ocoso_csar_25_1_201_0_0_0_prom_custom_values.yaml` is used for configuring the Prometheus parameters.
  - `ocoso_csar_25_1_201_0_0_0_alm_custom_values.yaml` is used for configuring the Alertmanager parameters.
  - `ocoso_csar_25_1_201_0_0_0_alr_config_custom_values.yaml` is used for configuring the Alert configuration parameters.

> **Note**
>
> The README doc contains the details to populate mandatory values in the `ocoso_csar_25_1_201_0_0_0_prom_custom_values.yaml` and `ocoso_csar_25_1_201_0_0_0_alm_custom_values.yaml` files. 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 packages are downloaded.
2. Unzip and TAR utilities are installed.
3. Docker or Podman (v5.4.0) is installed, and you must be able to run the docker or podman commands.
4. Minimum supported Helm version (v3.15.2) 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 repository server:

Table 2-1 Images in the Registry on the Repository Server

Image Name	Path in Repository with version
OSO_ALERTMANAGER_IMAGE	`occne.io/oso/alertmanager:`v0.28.0
OSO_PROMETHEUS_IMAGE	`occne.io/oso/prometheus:`v3.2.0
OSO_CONFIGMAPRELOAD_IMAGE	`occne.io/oso/configmapreload:`v0.15.0
OSO_ALERT_CONFIG_IMAGE	`occne.io/occne/oso_alert_config:`25.1.201
OSO_HELM_TEST_IMAGE	`occne.io/occne/oso_helm_test:`25.1.201
OSO_SNAPSHOT_IMAGE	`occne.io/occne/oso_snapshot:`25.1.201

All above images are packed into tar format and will be present in OSO CSAR under (Artifacts/Images) folder except for `oso_alert_config` which is present in `ocoso_alr_config_csar_25_1_201.zip`. 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> <registry-address>:<port>/image-url
$ docker/podman push <registry-address>:<port>/image-url
```

8. OSO can now be installed in one of the following ways:
   - without the snapshot utility feature (disabled): proceed as per existing procedure and resize the PVC storage.
   - with the snapshot utility feature (enabled): adjust the size of the PVC storage to fit the snapshot data. Otherwise, the Prometheus pod may crash. OSO recommends the users to configure thrice the existing storage capacity.
     If this feature is enabled, users experience variations in CPU and RAM usage. It is recommended that each NF should benchmark these figures and obtain them from the respective NFs.

     For example, if the PVC storage size required at user environment for OSO Prometheus is 50GB then in snapshot enabled OSO deployment, the storage configuration is given as 150GB (50GB*3). This value is defined in `PROM-PV-SIZE` for the required size.

     ```
     persistentVolume:
     enabled: true
     ...
     ...
     mountPath: /data
     size: PROM-PV-SIZE
     storageClass: "PROM-STORAGE-CLASS"
     ```

     Use the `web.enable-admin-api` parameter to enable the snapshot utility feature. For more information about this parameter, see Prometheus Configuration Parameters.

> **Note**
>
> Tables Table 2-2 and Table 2-3 in this section offer a comprehensive list of software necessary for the proper functioning of during deployment. However, these tables are indicative, and the software used can vary based on the customer's specific requirements and solution.

The `Software Requirement` column in Table 2-2 and Table 2-3 tables indicates one of the following:

- Mandatory: Absolutely essential; the software cannot function without it.
- Recommended: Suggested for optimal performance or best practices but not strictly necessary.
- Conditional: Required only under specific conditions or configurations.
- Optional: Not essential; can be included based on specific use cases or preferences.

Table 2-2 Preinstalled Software Versions

Software	25.1.2xx	25.1.1xx	24.3.x	Software Requirement	Usage Description
Helm	3.16.2	3.15.2	3.15.2	Mandatory	Helm, a package manager, simplifies deploying and managing NFs on Kubernetes with reusable, versioned charts for easy automation and scaling. **Impact**: Preinstallation is required. Without this capability, management of NF versions and configurations becomes time-consuming and error-prone, impacting deployment consistency.
Kubernetes	1.32.0	1.31.1	1.30.1	Mandatory	Kubernetes orchestrates scalable, automated NF deployments for high availability and efficient resource utilization. **Impact**: Preinstallation is required. Without orchestration capabilities, deploying and managing network functions (NFs) can become complex, leading to inefficient resource utilization and potential downtime.
Podman	5.4.0	5.4.0	5.4.0	Recommended	Podman is a part of Oracle Linux. It manages and runs containerized NFs without requiring a daemon, offering flexibility and compatibility with Kubernetes. **Impact**: Preinstallation is required. Without efficient container management, the development and deployment of NFs could become cumbersome, impacting agility.

To check the versions for the preinstalled software in the cloud native environment, run the following commands:

```
kubectl version
```

```
helm version
```

```
podman version
```

If you are deploying in a cloud native environment, these following additional software are to be installed before installing .

Table 2-3 Additional Software Versions

Software	25.1.2xx	25.1.1xx	24.3.x	Software Requirement	Usage Description
AlertManager	0.28.0	0.27.0	0.27.0	Recommended	Alertmanager is a component that works in conjunction with Prometheus to manage and dispatch alerts. It handles the routing and notification of alerts to various receivers. **Impact:** Not implementing alerting mechanisms can lead to delayed responses to critical issues, potentially resulting in service outages or degraded performance.
Configmapreload	0.15.0	0.14.0	0.13.0	Mandatory	Configmapreload is a simple binary to trigger a reload when Kubernetes ConfigMaps or Secrets, mounted into pods, are updated. It watches mounted volume dirs and notifies the target process that the config map has been changed. **Impact:** Not employing this solution would impact downtime as pods would need to restart everytime a configmap is updated.
Prometheus	3.2.0	2.52.0	2.52.0	Mandatory	Prometheus is a popular open source monitoring and alerting toolkit. It collects and stores metrics from various sources and allows for alerting and querying. **Impact:** Not employing this monitoring solution could result in a lack of visibility into NF performance, making it difficult to troubleshoot issues and optimize resource usage.

2.2 Resource Utilization

The use of resources such as CPU and RAM by the OSO services are constrained to ensure that the CPU and RAM do not consume excess resources that could be used by other applications.

During the deployment of the services, each service gets an initial CPU and RAM allocation. Each service is allowed to consume the resources to a specified upper limit while it continues to run. The initial allocation limit and the upper limit are set to the same value for services where the resource consumption limit remains the same as the initial allocation or in a case where increasing the CPU or RAM limits underneath a running application can cause service disruption.

> **Note**
>
> Each NF is supposed to benchmark their application against OSO services and set the RAM, CPU, and PVC values accordingly. These initial resource request is set by OSO team as a placeholder and it is not recommended for any NF.

The resource requests and limits are as follows:

Table 2-4 CPU, RAM, and Ephemeral Resource Requests and Limits

Service	CPU Initial Request (m)	CPU Limit (m)	RAM Initial Request (Mi)	RAM Limit (Mi)	PVC (Gb)	Ephemeral Storage Request (Gi)	Ephemeral Storage Limit (Gi)	Instances
Prometheus	2000	4000	16384 (16 Gb)	32768 (32 Gb)	50	4	6	1
Prometheus (with snapshot utility enabled)	4000	8000	32768 (32 Gb)	65536 (64 Gb)	150	8	12	1
Prometheus AlertManager	20	20	64	64	NA	1	1	2

The overall observability services resource usage varies on each worker node. The observability services listed in the table are distributed evenly across all worker nodes in the Kubernetes cluster.

2.3 Downloading the OSO package

To download the OSO package from My Oracle Support, perform the following procedure:

1. Log in to My Oracle Support using your login credentials.
2. Click the **Patches & Updates** tab to locate the patch.
3. In the **Patch Search** console, click the **Product or Family (Advanced)** option.
4. In the **Product** field, enter `Oracle Communications Cloud Native Core - 5G` and select the product from the Product drop-down list.
5. From the **Release** drop-down list, select **Oracle Communications Cloud Native Core Network Repository Function *<release_number>***.
   Where, *<release_number>* indicates the required release number of OSO.
6. Click **Search**.
   The Patch Advanced Search Results list appears.
7. Select the required patch from the list.
   The Patch Details window appears.
8. Click **Download**.
   The File Download window appears.
9. Click the `<p********_<release_number>_Tekelec>.zip` file to download the release package.

Where, <p********> is the MOS patch number and <release_number> is the release number of OSO.

2.4 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 files from My Oracle Support (MOS).
2. Extract the CSAR zip files:

```
$ unzip <OSO CSAR package>
$ unzip -d ./oso_csar/ ocoso_csar_<25.1.201 version>.zip
$ unzip -d ./oso_alert_config_csar/ ocoso_alr_config_csar_<25.1.201
version>.zip
```

3. Upload all the artifacts present in the `Artifacts/Images` folder to the configured repository.
   The following structure indicates the CSAR file containing the Prometheus and alertmanager files.

```
├── Artifacts
│   ├── ChangeLog.txt
│   ├── Images
│   │   ├── alertmanager-0.28.0.tar
│   │   ├── configmapreload-0.15.0.tar
│   │   ├── oso-helm-test-25.1.201.tar
│   │   ├── oso-snapshot-25.1.201.tar
│   │   └── prometheus-3.2.0.tar
│   ├── Licenses
│   ├── Scripts
│   │   ├── ocoso_csar_25_1_201_0_0_0_alertmanager_charts.tgz
│   │   ├── ocoso_csar_25_1_201_0_0_0_alm_custom_values.yaml
│   │   ├── ocoso_csar_25_1_201_0_0_0_prometheus_charts.tgz
│   │   ├── ocoso_csar_25_1_201_0_0_0_prom_custom_values.yaml
│   │   └── OSO_README.md
│   └── Tests
├── Definitions
│   ├── etsi_nfv_sol001_vnfd_2_5_1_types.yaml
│   ├── oso_vnfd.yml
│   └── helm_extensions_1_0.yaml
├── ocoso_csar_25_1_201_0_0_0.zip
├── oso_vnfd.mf
└── TOSCA-Metadata
    └── TOSCA.meta
```

The following structure indicates the CSAR file containing the alert configuration files.

```
.
├── Artifacts
│   ├── ChangeLog.txt
│   ├── Images
```

```
│   │   └── oso-alert-config-25.1.201.tar
│   ├── Licenses
│   ├── Scripts
│   │   ├── ocoso_alr_config_csar_25_1_201_0_0_0_alert_config_charts.tgz
│   │   ├──
ocoso_alr_config_csar_25_1_201_0_0_0_alert_config_custom_values.yaml
│   │   └── OSO_ALR_CONFIG_README.md
│   └── Tests
├── Definitions
│   ├── etsi_nfv_sol001_vnfd_2_5_1_types.yaml
│   ├── oso_alr_config_vnfd.yml
│   └── helm_extensions_1_0.yaml
├── ocoso_alr_config_csar_25_1_201_0_0_0.zip
├── oso_alr_config_vnfd.mf
└── TOSCA-Metadata
    └── TOSCA.meta
```

For more information about the artifacts, see the Prerequisites section.

4. From 25.1.200 onwards, the `custom_values.yaml` file is split into three different files:
   - `ocoso_csar_25_1_201_0_0_0_prom_custom_values.yaml` is used for configuring the Prometheus parameters
   - `ocoso_csar_25_1_201_0_0_0_alm_custom_values.yaml` is used for configuring the Alertmanager parameters.
   - `ocoso_csar_25_1_201_0_0_0_alert_config_custom_values.yaml` is used for configuring the alerts.

   These files are required to be populated and installed separately. Update the respective `custom_values.yaml` files inside respective `Artifacts/Images` directory with the required values as mentioned in OSO Configuration Parameters.
5. Enable IPv6 Dualstack.
   a. **To enable IPv6 DualStack in Prometheus**
      - Enable IPv6 Dualstack in the `ocoso_csar_25_1_201_0_0_0_prom_custom_values.yaml` file. 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
        ```

      - Change the service type from ClusterIP to LoadBalancer to assign IPv6.
      - Save the file and proceed with the normal installation.

   b. **To enable IPv6 DualStack in AlertManager** Replace "ENABLE_DUAL_STACK" with "true" in below section inside the

`ocoso_csar_25_1_201_0_0_0_alm_custom_values.yaml` file before installing alertmanager.

```
# ip dual stack
  ipDualStack:
    enabled: ENABLE_DUAL_STACK
    ipFamilies: ["IPv6", "IPv4"]
    ipFamilyPolicy: "PreferDualStack"
```

6. Save the file and proceed with the normal installation.
7. Install OSO using the following steps:
   a. Extract the Helm charts tgz files which are available in the `Artifacts/Scripts` directory and populate the three `custom_values.yaml` files.

```
$ cd Artifacts/Scripts
$ tar -xvzf ocoso_csar_25_1_201_0_0_0_alertmanager_charts.tgz
$ tar -xvzf ocoso_csar_25_1_201_0_0_0_prometheus_charts.tgz
$ tar -xvzf ocoso_csar_25_1_201_0_0_0_alert_config_charts.tgz
```

   b. Install Prometheus and Alertmanager using Helm charts provided and updated `ocoso_csar_25_1_201_0_0_0_prom_custom_values.yaml` and `ocoso_csar_25_1_201_0_0_0_alm_custom_values.yaml` file with the following commands. To install alert-config feature, see Automated Configuration of NF Alerts. Use the following command for installation using Helm, if custom labels are given:

```
$ kubectl create namespace <deployment-namespace-name>
$ helm --namespace=<deployment-namespace-name> install <oso-prom-
release-name> -f <ocoso_csar_25_1_201_0_0_0_prom_custom_values.yaml> ./
ocoso_csar_25_1_201_0_0_0_prometheus_charts.tgz
$ helm --namespace=<deployment-namespace-name> install <oso-
alertmanager-release-name> -f
<ocoso_csar_25_1_201_0_0_0_alm_custom_values.yaml> ./
ocoso_csar_25_1_201_0_0_0_alertmanager_charts.tgz
```

For example:

```
$ kubectl create namespace <oso-namespace>
# Example for Prometheus:
$ helm --namespace=<oso-namespace> install <oso-prom-release-name> -f
ocoso_csar_25_1_201_0_0_0_prom_custom_values.yaml
        ./ocoso_csar_25_1_201_0_0_0_prometheus_charts.tgz
# Similarly for Alertmanager
$ helm --namespace=<oso-namespace> install <oso-alertmanager-release-
name> -f ocoso_csar_25_1_201_0_0_0_alm_custom_values.yaml
        ./ocoso_csar_25_1_201_0_0_0_alertmanager_charts.tgz
```

```
$ kubectl create namespace ns1
## Prometheus
$helm install -f ocoso_csar_25_1_201_0_0_0_prom_custom_values.yaml --
namespace=ns1 --name-template=oso-p ./
```

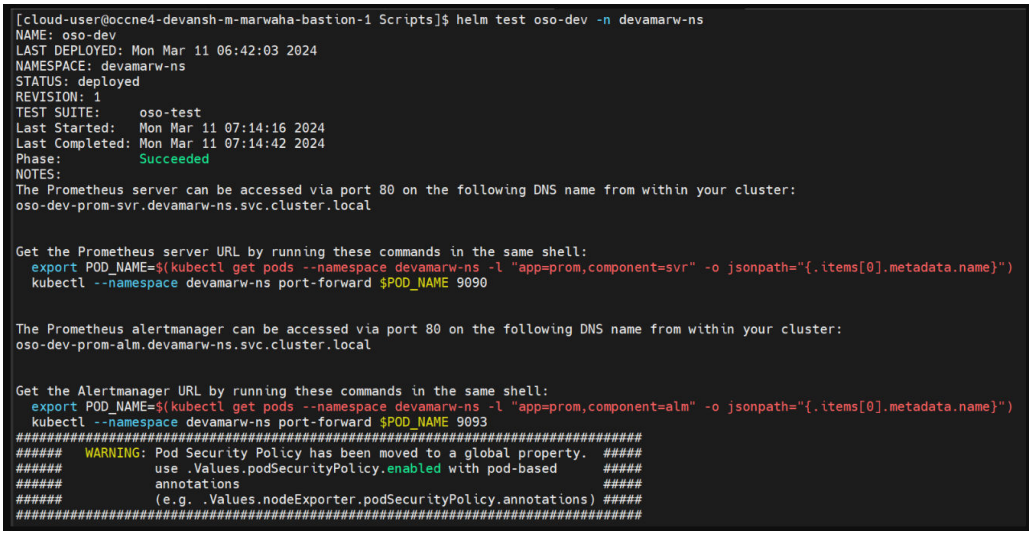
```
ocoso_csar_25_1_201_0_0_0_prometheus_charts.tgz
##  Alertmanager
$ helm install -f ocoso_csar_25_1_201_0_0_0_alm_custom_values.yaml --
namespace=ns1 --name-template=oso-a ./
ocoso_csar_25_1_201_0_0_0_alertmanager_charts.tgz
```

c. Run the following command to perform a Helm test. Populate the values in the `ocoso_csar_25_1_201_0_0_0_prom_custom_values.yaml` and `ocoso_csar_25_1_201_0_0_0_alm_custom_values.yaml` files to run the Helm test. Run the Helm test separately for each Alertmanager and Prometheus.

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



2.4.1 Automated Configuration of NF Alerts

This section explains the process of automating the NF alerts configuration in OSO after deploying all the relevant NFs and OSO components. The process manages the alert rule definitions and dynamically updates the OSO Prometheus configmap as part of the Helm chart installation or upgrade.

1. **Prerequisites**
   a. Install the respective NF, CNC Console, cnDBTier, and OSO in their designated namespace.
   b. The existing structure of the `nf-alertrule.yaml` file remains unchanged. Operators must prepare the alert rule file(s) and place them in a directory on the server, which will be provided as input to the new Helm chart.
2. **Existing Functionality**
   NF Alerts are configured in Prometheus manually post OSO deployment by performing the following steps:

```
kubectl get configmaps occne-prometheus-server -o yaml -n <oso-namespace>
> /tmp/tempConfig.yaml
$ sed -i '/etc\/config\/alertscncc/d' /tmp/tempConfig.yaml
$ sed -i '/rule_files:/a\ \- /etc/config/alertscncc' /tmp/tempConfig.yaml
$ kubectl replace configmap occne-prometheus-server -f /tmp/tempConfig.yaml
$ kubectl patch configmap occne-prometheus-server -n occne-infra --type
merge --patch "$(cat ~/occncc_alertrules_<version>.yaml)"
```

3. **Alert Automation**
   a. **Installing Alert Automation feature using Helm**
      i. Extract the Helm charts tgz file and available in the `Artifacts/Scripts` directory.

```
$ cd Artifacts/Scripts
$ tar -xvzf ocoso_csar_25_1_200_0_0_0_alert_config_charts.tgz
```

> **Note**
>
> Deploy the `oso-alr-config` Helm chart after OSO Prometheus and Alertmanager are installed using the `helm install` command.

      ii. Provide the appropriate values in the `ocoso_csar_25_1_201_0_0_0_alert_config_custom_values.yaml` file for `oso-alr-config` configmap and proceed with the installation of the alert automation feature. The chart can accept one or more alert rule files during deployment.
         - Install the alert configuration feature by providing one alert-rule file:

```
helm -n <oso-namespace> install <oso-alr-config-feature-name> ./
oso-alr-config -f
ocoso_csar_25_1_200_0_0_alert_config_custom_values.yaml -f
path/to/nf_alertrules.yaml
```

         - Install the alert configuration feature by providing multiple alert-rule files:

```
helm -n <oso-namespace> install <oso-alr-config-feature-name> ./
oso-alr-config -f
ocoso_csar_25_1_200_0_0_alert_config_custom_values.yaml -f
path/to/nf1_alertrules.yaml -f path/to/nf2_alertrules.yaml
```

iii. Run the following command to verify if `oso-alr-config` configmap is created by Helm.

```
$ kubectl -n <oso-namespace> get configmap
```

Example:

```
kubectl -n oso get cm
```

Sample output

```
NAME DATA AGE
oso-alr-config-cm 2 2d21h
oso-prom-svr 8 2d21h
```

iv. Verify if the alerts are successfully loaded in the Prometheus GUI. For more information, see Verifying Installation section to understand how to access OSO Prometheus GUI and how to verify alerts.

v. If no alert rule file is provided during installation, the script will not perform any actions, but the `oso-alr-config` Helm chart will still be installed. It is strongly recommended to supply at least one alert rule file, as it is required for configuration.

> **Note**
>
> The deployment fails in any of the following cases:
>
> - If `OSO-PROM-CONFIGMAP-NAME` does not exist (Helm deployment fails before triggering script).
> - If there are validation errors.
> - If K8s API returns error.

b. **Adding more alerts using Helm Upgrade**

- Prepare the NF alert rule file and trigger a Helm upgrade for the `oso-alr-config` chart by providing the alert rule file as input.
- The deployment fails in any of the following cases:

> **Note**
>
> – If `OSO-PROM-CONFIGMAP-NAME` or `oso-alr-config-cm` does not exist.
> – If there are validation errors.
> – If K8s API returns error.

- The chart can accept one or more alert rule files during deployment. Run the following command to provide multiple files:

  ```
  Using Multiple -f Flags
  helm upgrade oso-alr-config oso-alr-config/ -f
  <ocoso_csar_25_1_200_0_0_alert_config_custom_values.yaml> -f
  <path/to/nf1_alertrules.yaml> -f <path/to/nf2_alertrules.yaml> -n
  <oso-namespace>
  ```

  > **Note**
  >
  > It is recommended to provide all the alert files together like all NFs, cnDBTier, and CNC Console).

- `OSO_ALERTRULE_CONFIGMAP` will be created by Helm and will be accessible to user using the following command:

  ```
  $ kubectl -n <oso-namespace> get configmap
  ```

  For example, `kubectl -n oso get cm`

  Sample output:

  ```
  NAME DATA AGE
  oso-alr-config-cm 2 2d21h
  oso-prom-svr 8 2d21h
  ```

- After successfully verifying the upgrade, Route to the OSO Prometheus GUI to verify if the Alerts are successfully loaded, For more information about accessing Prometheus GUI and verifying the alerts, see Verifying Installation.

c. **Unpatching or removing the alert automation feature**

- Clear the alerts from the OSO Prometheus ConfigMap and Prometheus GUI by using an empty alert rule file will be provided during the Helm upgrade. However, the references are retained under `rule_files` "`/etc/config/alertscncc`" and alert rules will be empty "`alertscncc: { }`"

  > **Note**
  >
  > It is recommended to unpatch all alerts together. This can be achieved by providing all the empty alert files at once, which restores the alerts section in the OSO deployment to its original state.

- Prepare empty alert files (`empty_nf_alertrules.yaml`) for all the respective NFs that should be removed from the Prometheus GUI.
  For example, CNC Console alerts should not be listed under Prometheus GUI.

  ```
  apiVersion: v1
  data:
  ```

```
  alertscncc: |
    {}
```

- Run a helm upgrade (for more information, see step 3.b) and pass these empty files.
  The following example, indicates when only 1 empty file of CNC Console is given to display the effect of this empty file.

  Updated Prometheus ConfigMap (`oso-prom-svr configMap` output):

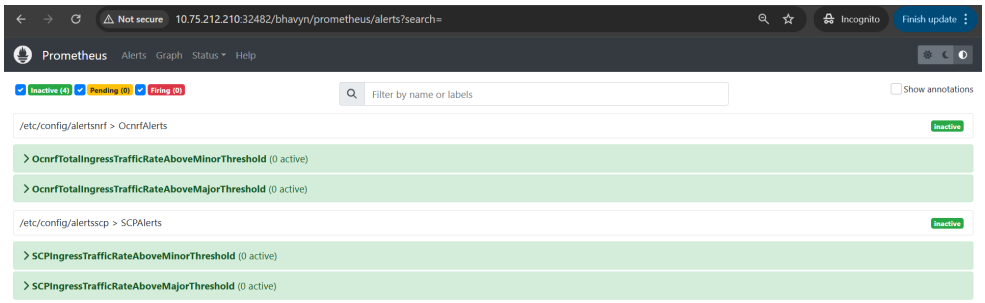
```
kubectl get configmap/oso-prom-svr -n default -oyaml
apiVersion: v1
data:
  alerting_rules.yml: |
    {}
  alerts: |
    {}
  alertscncc: |
    {}
  alertsnrf: |
    groups:
    - name: OcnrfAlerts
      rules:
      - alert: OcnrfTotalIngressTrafficRateAboveMinorThreshold
        annotations:
          description: 'Ingress traffic Rate is above the
configured minor threshold i.e. 800 requests per second (current
value is: {{ $value }})'
          summary: 'timestamp: {{ with query "time()" }}{{ . |
first | value | humanizeTimestamp }}{{ end }}: Traffic Rate is
above 80 Percent of Max requests per second(1000)'
        expr:
sum(rate(oc_ingressgateway_http_requests_total{app_kubernetes_io_nam
e="ingressgateway",kubernetes_namespace="ocnrf"}[2m])) by
(kubernetes_namespace) >= 800 < 900
        labels:
          severity: minor
          oid: "1.3.6.1.4.1.323.5.3.36.1.2.7001"
          kubernetes_namespace:
' {{ $labels.kubernetes_namespace }} '
      - alert: OcnrfTotalIngressTrafficRateAboveMajorThreshold
        annotations:
          description: 'Ingress traffic Rate is above the
configured major threshold i.e. 900 requests per second (current
value is: {{ $value }})'
          summary: 'timestamp: {{ with query "time()" }}{{ . |
first | value | humanizeTimestamp }}{{ end }}: Traffic Rate is
above 90 Percent of Max requests per second(1000)'
        expr:
sum(rate(oc_ingressgateway_http_requests_total{app_kubernetes_io_nam
e="ingressgateway",kubernetes_namespace="ocnrf"}[2m])) by
(kubernetes_namespace) >= 900 < 950
        labels:
          severity: major
          oid: "1.3.6.1.4.1.323.5.3.36.1.2.7002"
          kubernetes_namespace:
```

```
' {{ $labels.kubernetes_namespace }} '
  alertsscp: |
    groups:
    - name: SCPAlerts
      rules:
      - alert: SCPIngressTrafficRateAboveMinorThreshold
        annotations:
          description: 'Ingress Traffic Rate at locality:
"{{$labels.ocscp_locality}}"  is above minor threshold (i.e. 2380
mps)'
          summary: 'namespace: {{$labels.kubernetes_namespace}},
podname: {{$labels.kubernetes_pod_name}},scpfqdn:
{{$labels.ocscp_fqdn}},scpauthority:{{$labels.ocscp_authority}},
timestamp: {{ with query "time()" }}{{ . | first | value |
humanizeTimestamp }}{{ end }}: Current Ingress Traffic Rate is
{{ $value | printf "%.2f" }} mps which is above 70 Percent of Max
MPS(3400)'
        expr:
sum(rate(ocscp_metric_http_rx_req_total{app_kubernetes_io_name="scp-
worker"}[2m])) by
(kubernetes_namespace,ocscp_locality,kubernetes_pod_name,ocscp_fqdn,
ocscp_authority) >= 2380 < 2720
        for: 1m
        labels:
          severity: minor
          alertname: "SCPIngressTrafficRateAboveMinorThreshold"
          oid: "1.3.6.1.4.1.323.5.3.35.1.2.7001"
          namespace: '{{$labels.kubernetes_namespace}}'
          podname: '{{$labels.kubernetes_pod_name}}'
          vendor: oracle
      - alert: SCPIngressTrafficRateAboveMajorThreshold
        annotations:
          timestamp: ' {{ with query "time()" }}{{ . | first |
value | humanizeTimestamp }}{{ end }} '
          description: 'Ingress Traffic Rate at locality:
{{$labels.ocscp_locality}} is above major threshold (i.e. 2720 mps)'
          summary: 'namespace: {{$labels.kubernetes_namespace}},
podname: {{$labels.kubernetes_pod_name}},scpfqdn:
{{$labels.ocscp_fqdn}},scpauthority:{{$labels.ocscp_authority}},
timestamp: {{ with query "time()" }}{{ . | first | value |
humanizeTimestamp }}{{ end }}: Current Ingress Traffic Rate is
{{ $value | printf "%.2f" }} mps which is above 80 Percent of Max
MPS(3400)'
        expr:
sum(rate(ocscp_metric_http_rx_req_total{app_kubernetes_io_name="scp-
worker"}[2m])) by
(kubernetes_namespace,ocscp_locality,kubernetes_pod_name,ocscp_fqdn,
ocscp_authority)  >= 2720 < 3230
        for: 1m
        labels:
          severity: major
          alertname: "SCPIngressTrafficRateAboveMajorThreshold"
          oid: "1.3.6.1.4.1.323.5.3.35.1.2.7001"
          namespace: '{{$labels.kubernetes_namespace}}'
          podname: '{{$labels.kubernetes_pod_name}}'
          vendor: oracle
```

```
  allow-snippet-annotations: "false"
  prometheus.yml: |
    global:
      evaluation_interval: 1m
      scrape_interval: 1m
      scrape_timeout: 30s
    rule_files:
    - /etc/config/alertsnrf
    - /etc/config/recording_rules.yml
    - /etc/config/alerts
    - /etc/config/alerting_rules.yml
    - /etc/config/alertsscp
    - /etc/config/alertscncc
    - /etc/config/rules
......
......
......
```

Screenshot of Prometheus GUI without CNC Console alerts:

Figure 2-2 Prometheus GUI with No CNC Console Alert Files



d. Helm Uninstall

i. Run the following command to uninstall `oso-alr-config` Helm chart.

```
helm uninstall oso-alr-config -n <oso-namespace>
```

During `oso-alr-config` Helm chart uninstallation, no hooks are executed. As a result, the OSO Prometheus configmapMap and its alert configurations remain intact, while all other `oso-alr-config` resources are deleted.

ii. In the event of a failure during an upgrade followed by an uninstall, the failed jobs may persist. In such cases, you must manually clear all jobs using the following command:

- Run the following command to retrieve the jobs:

```
#Get the jobs
$ kubectl get jobs -n <oso-namespace>
```

Sample output:

```
NAME                                 COMPLETIONS   DURATION   AGE
oso-alr-config-hook-post-install     0/1           138m       138m
```

- Run the following command to delete the jobs:

```
# Delete the jobs
$ kubectl delete job oso-alr-config-hook-post-install
```

Sample output:

```
job.batch "oso-alr-config-hook-post-install" deleted
```

2.5 Postinstallation Tasks

This section explains the postinstallation tasks for OSO.

2.5.1 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 2-3 Prometheus GUI

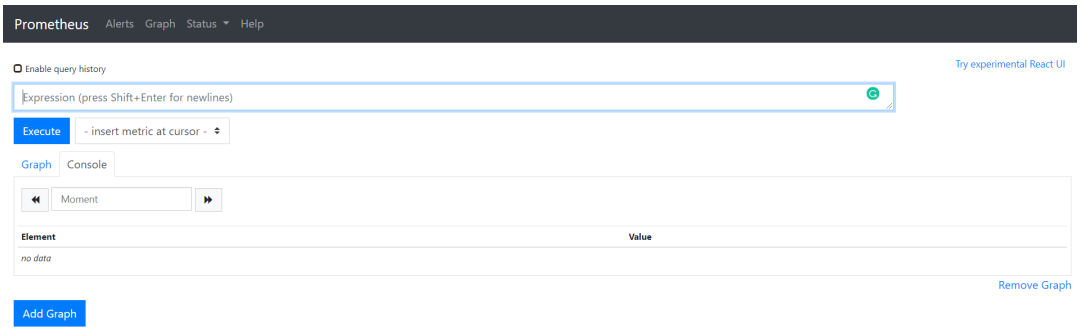
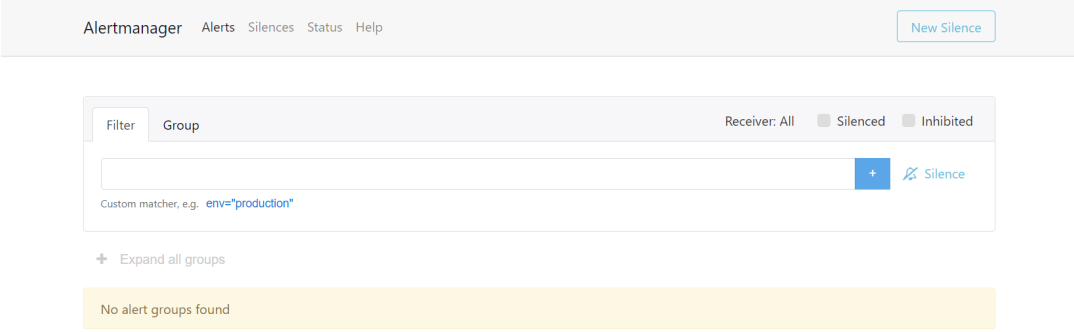


Figure 2-4 Alert Manager GUI

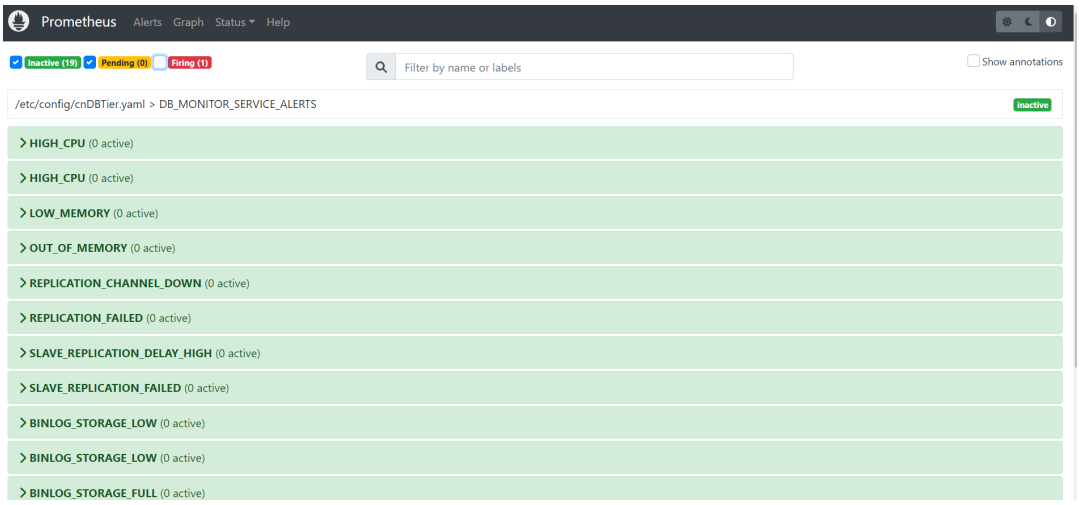


5. On the Prometheus GUI, click **Alerts** to verify that all the alerts (NF Alerts) are visible.

> **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.

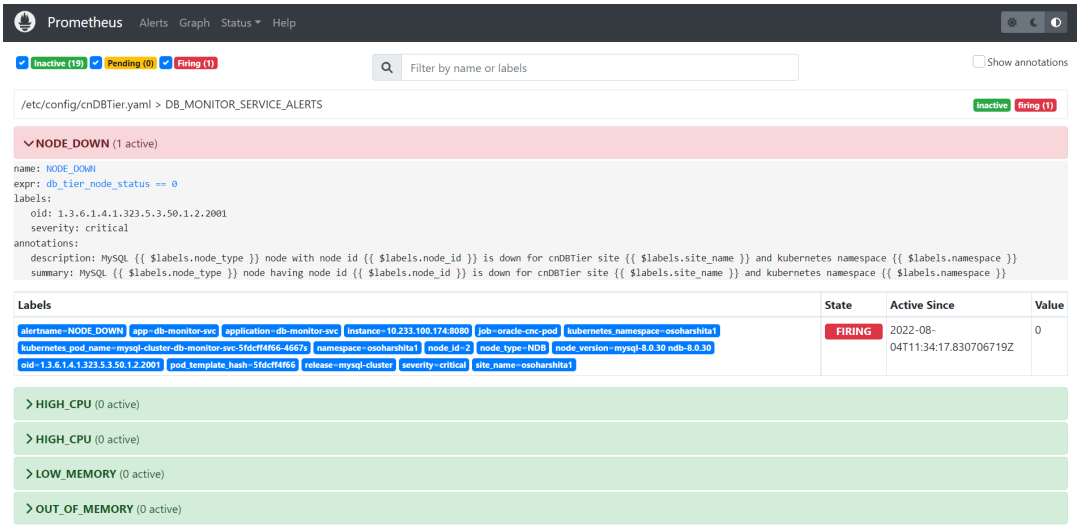
The following image displays the alerts for cnDBTier:

Figure 2-5 Prometheus GUI



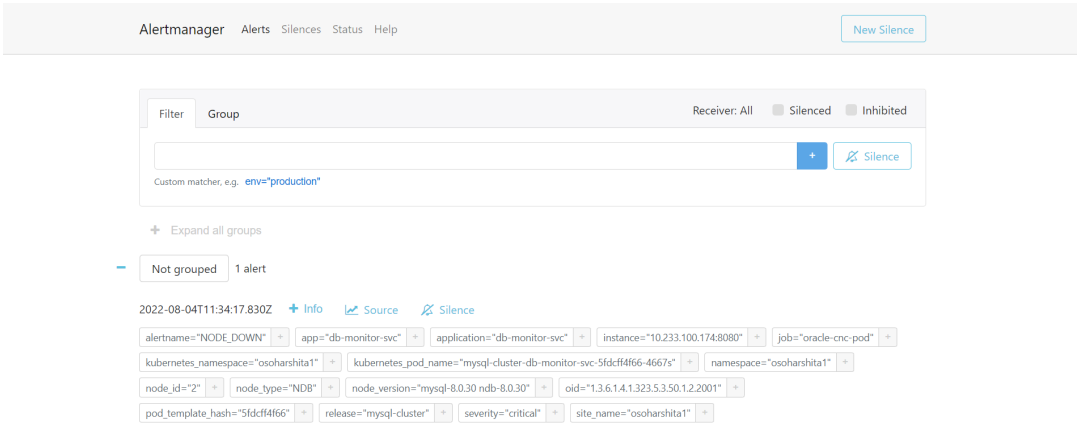
After alerts are raised, the GUI must display the triggered alerts as shown in the following image:

Figure 2-6 Prometheus GUI - Alerts



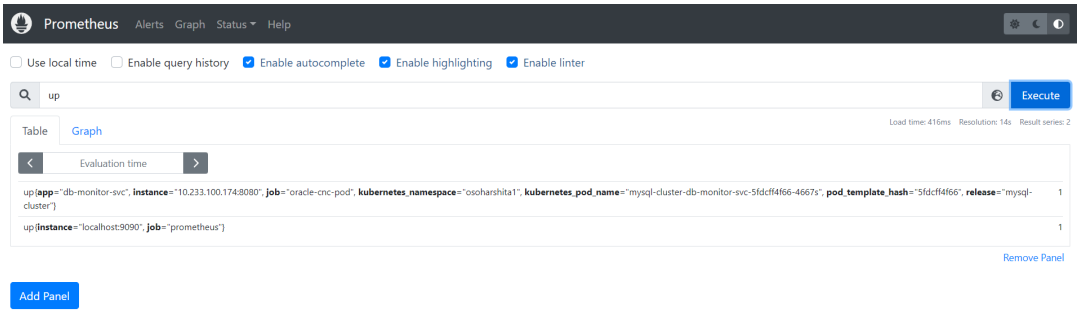
6. Select **Alerts** tab on the Alertmanager GUI to view the triggered alerts as shown in the following image:

Figure 2-7 Alertmanager - Alerts



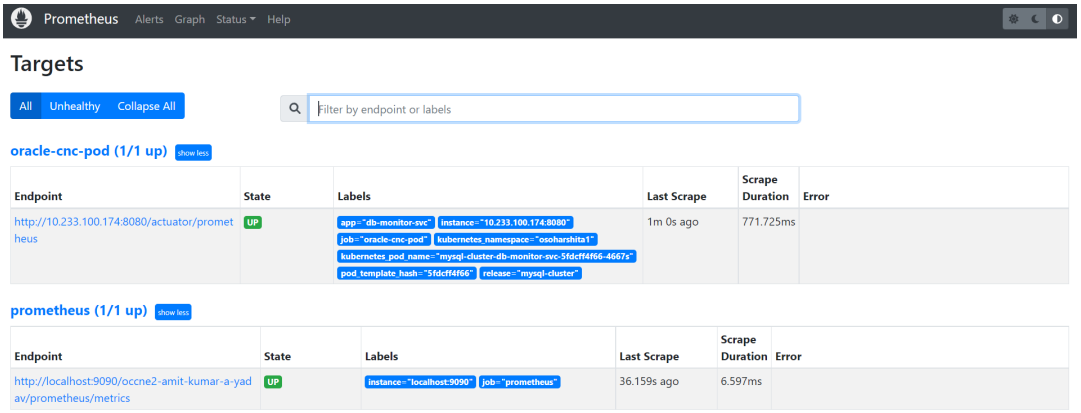
7. On the Prometheus GUI, click **Graph** to verify if the expected metrics (example, NF Metrics) are displayed. The following image displays a sample Prometheus Graph with metrics:

Figure 2-8 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 2-9 Prometheus Target



> **Note**
>
> Post installation, Prometheus can be vertically scaled to add more resources to an existing Prometheus server for load balancing. For more information about, Prometheus vertical scaling, see Prometheus Vertical Scaling.

3
Customizing OSO

This chapter provides information about customizing Oracle Communications Cloud Native Core, 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>` files.

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 • prom-svr
Alertmanager(prom-alm)	• prom-alm • prom-alm-cm-r

You can use the following values in the `ocoso_csar_25_1_201_0_0_0_prom_custom_values.yaml` and `ocoso_csar_25_1_201_0_0_0_alm_custom_values.yaml` files 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 `ocoso_csar_25_1_201_0_0_0_prom_custom_values.yaml` and `ocoso_csar_25_1_201_0_0_0_alm_custom_values.yaml` files:
   a. **Global Labels(allResources)**: These labels are attached to all the Kubernetes resources. This is depicted as TYPE1 in the following sample `ocoso_csar_25_1_201_0_0_0_prom_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 `ocoso_csar_25_1_201_0_0_0_prom_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 `ocoso_csar_25_1_201_0_0_0_prom_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.

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:

```
OSO_ALERTMANAGER_IMAGE=occne.io/occne/alertmanager
OSO_PROMETHEUS_IMAGE=occne.io/occne/prometheus
OSO_CONFIGMAPRELOAD_IMAGE=occne.io/occne/configmapreload
OSO_HELM_TEST_IMAGE=occne.io/occne/oso_helm_test
OSO_ALERT_CONFIG_IMAGE=occne.io/occne/oso_alert_config
OSO_SNAPSHOT_IMAGE=occne.io/occne/oso_snapshot
```

Prerequisites

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

3.2.1 Prometheus Configuration Parameters

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

Configure `ocoso_csar_25_1_201_0_0_0_prom_custom_values.yaml` with the specific values.

Table 3-2 OSO Configuration Parameter

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

Table 3-2 (Cont.) OSO Configuration Parameter

Parameter	Description	Details
CLUSTER-NAME	This is a mandatory parameter. Indicates the name of the cluster where OSO is deployed. This value is required in multiple places in the custom values.yaml file. Example: clustername: oso-cluster	**Data Type:** String **Default Value:** NA **Range:** NA

USE-DEFAULT-SVCACCOUNT-PROM	This is a mandatory parameter.
	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.
	Example:
	`serviceAccounts:`
	`  prometheus:`
	`  create: false`
		Data Type: Boolean
		Default Value: false
		Range: true, false

serviceAccountNamePromSvr	This is a mandatory parameter.
	Indicates the ServiceAccount that is to be used by the Prometheus resource.
	Users must create ServiceAccount with the required namespace access defined along with the Role or RoleBinding for the Prometheus service.
	Example:
	`# Service account for Prometheus`
	`  serviceAccountNamePromSvr: "oso-prom"`
	If you do not have a ServiceAccount, then set the USE-DEFAULT-SVCACCOUNT-PROM parameter to true.
		Data Type: String
		Default Value:""
		Range: NA

PROMETHEUS-REPO-ADDRESS	This is a mandatory parameter. Indicates the docker image repo URL for Prometheus location. Example: `Bastion:5000/occne.io/oso/prometheus`	**Data Type:** String **Default Value:** NA **Range:** NA

Table 3-2 (Cont.) OSO Configuration Parameter

Parameter	Description	Details
HELM-TEST-IMAGE-REPO-ADDRESS	This is a mandatory parameter. Indicates the repo URL of the Helm test image location. Example: `Bastion:5000:occne.io/occne/oso_helm_test`	**Data Type:** String **Default Value:** NA **Range:** NA
CONFIGMAP-RELOAD-REPO-ADDRESS	This is a mandatory parameter. Indicates the docker image repository URL for configmap-reload location. Example: `Bastion:5000/occne.io/oso/configmapreload`	**Data Type:** String **Default Value:** NA **Range:** NA

server:
.....
##
namespaces to monitor(instead of monitoring all - clusterwide).
 	
##Needed if you want to run without Cluster-admin privileges.
 	
namespaces:
- NAMESPACE
	This is a mandatory parameter.
	If clusterrole is used, comment the namespaces section. If role is used, then give the namespace where OSO is deployed.
	Example:
	
	`server:`
	`.....`
	`## namespaces to monitor(instead of monitoring all - clusterwide).`
	`## Needed if you want to run without Cluster-admin privileges.`
	`## namespaces:`
	`## - NAMESPACE`
		Data Type: String
		Default Value: NA
		Range: NA

Table 3-2 (Cont.) OSO Configuration Parameter

Parameter	Description	Details

BEARER_TOKEN	This is a conditional parameter.
	Set the Authorization header on every remote write request with the configured bearer token. In this case, it is Cortex Server.
	Example:
	
	`remote_write`
	`# bearer_token for cortex server to be configured`
	`bearer_token: eyJhbGciOiJIUzUxMiIsInR5cCI6IkpXVCJ9.`
		Data Type: String
		Default Value: NA
		Range: NA

STORAGE_RETENTION_SIZE_PROM (has two variants)	This is a mandatory parameter.
	Indicates the retention size for Prometheus persistence volume.
	It should be 80% of the PV size (6.8GB).
	Retention size will remain the same percentage of storage size in both snapshot enabled or disabled cases.
	Example:
	
	`If PV = 8GB`
	`With Snapshot enabled:`
	`PV size: 24GB`
	`Without Snapshot enabled:`
	`PV size: 8GB`
	`Retention in both cases: 6.4GB`
	
	`extraArgs:`
	
	`storage.tsdb.retention.size: 6.4GB`
		Data Type: String
		Default Value: 7.3 GB
		Range: NA

PROM-PV-SIZE	This is a mandatory parameter.
	Indicates the size for the persistence volume of Prometheus.
	Example:
	
	`persistentVolume:`
	`  enabled: true`
	`  size: 8Gi`
		Data Type: String
		Default Value: NA
		Range: NA

Table 3-2 (Cont.) OSO Configuration Parameter

Parameter	Description	Details

PROM-STORAGE-CLASS	This is a mandatory parameter.
	Indicates the storage class for Prometheus.
	Example:
	`persistentVolume:`
	`enabled: true`
	`......`
	`storageClass: standard`
		Data Type: String
		Default Value:standard
		Range: NA

PROM-DATA-RETENTION-DAYS	This is a mandatory parameter. Indicates the retention days for Prometheus persistence volume. Example: `retention: 7d`	**Data Type:** String **Default Value:** NA **Range:** NA
OSO_CORTEX_URL	This is a conditional parameter. Indicates the URL where Prometheus writes metrics data. Example: `http://<some-ip>:<some-port>/api/prom/push`	**Data Type:** String **Default Value:** NA **Range:** NA
OSO_REMOTE_WRITE_TIMEOUT	This is a conditional parameter. Indicates the maximum amount of time Prometheus waits for a response to the remote write request. Example: 30s	**Data Type:** String **Default Value:** NA **Range:** NA

Table 3-2 (Cont.) OSO Configuration Parameter

Parameter	Description	Details

`namespaces`	This is a conditional parameter.
	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.
	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.
	`- job_name: 'oracle-cnc-pod'`
	`kubernetes_sd_configs:`
	`- role: pod`
	`#namespaces:`
	`# names:`
	`# - ns1`
	`# - ns2`
	You can add more namespaces in the list as per your requirement.
		Data Type: String
		Default Value: NA
		Range: NA

`Prefix`
`Suffix`
	This is a conditional parameter.
	Indicates the prefix and suffix that will be added to K8s Resource names.
	Example:
	`global:`
	`k8Resource:`
	`container:`
	`prefix: oracle1`
	`suffix: oracle2`
	Note: Currently, only Containers are configured with prefix and suffix.
		Data Type: String
		Default Value: NA
		Range: NA

`allResources:`
`labels:`
`{}`
	This is a conditional parameter.
	Indicates the Global Labels (allResources) that will be attached to all Kubernetes resources.
	Example:
	`allResources:`
	`labels:`
	`env : production`
		Data Type: String
		Default Value: NA
		Range: NA

Table 3-2 (Cont.) OSO Configuration Parameter

Parameter	Description	Details

`lbServices: labels: {} nonlbServices: labels: {} lbDeployments: labels: {} nonlbDeployments: labels: {} lbStatefulSets: labels: {}`	This is a conditional parameter.
	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.
	Example:
	`lbServices:`
	`  labels:`
	`  key1 : value1`
	`nonlbServices:`
	`  labels:`
	`  key2 : value2`
	`lbDeployments:`
	`  labels:`
	`  key3 : value3`
	`nonlbDeployments:`
	`  labels:`
	`  key4 : value4`
	`lbStatefulSets:`
	`  labels:`
	`  key5 : value5`
		Data Type: String
		Default Value: NA
		Range: NA

`annotations: {}`	This is a conditional parameter.
	Specifies the custom annotations to be added to OSO deployments that are associated to disable the sidecar injections into OSO pods.
	Note: This parameter is mandatory if ASM is enabled.
	Example:
	`annotations:`
	`  - sidecar.istio.io/inject: "false"`
		Data Type: Boolean
		Default Value: false
		Range: true, false

Table 3-2 (Cont.) OSO Configuration Parameter

Parameter	Description	Details
alertmanagers: - kubernetes_sd_configs: - role: pod # namespace to be configured namespaces: names: {} relabel_configs: - source_labels: [__meta_kubernetes_namespace] # namespace to be configured regex: {}	This is a mandatory parameter. Indicates if this parameter is used to add namespaces for Alertmanager. Example: 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 **Default Value:** NA **Range:** NA

Table 3-2 (Cont.) OSO Configuration Parameter

Parameter	Description	Details

`extraFlags:`
`- web.enable-lifecycle`
`- web.enable-admin-api`
	This is a conditional parameter.
	Indicates that the `web.enable-admin-api` parameter controls access to the administrative HTTP API. This parameter is disabled by default.
	Uncomment the `web.enable-admin-api` parameter to enable it. If the parameter is enabled, it allows capturing snapshots in TSDB.
		Data Type: String
		Default Value: disabled
		Range:disabled, enabled

Table 3-2 (Cont.) OSO Configuration Parameter

Parameter	Description	Details
serverFiles: prometheus.yml: rule_files: .... # All the Alert-rule names given below MUST be verified by the user before attempting UPGRADE, the names has to match N-1 Alert-rule names in order to guarantee their preservation through Upgrade. - /etc/config/alertscncc - /etc/config/alertscnd	This is a conditional parameter. Indicates that the NF Alert-rule names are hard-coded and must be verified by the user during upgrade to make sure the names are correct before upgrade. This helps to preserve the NF alert-rules in Prometheus GUI. If the names are different, then an update is required.	**Data Type:** String **Default Value:** NA **Range:**NA

Table 3-2 (Cont.) OSO Configuration Parameter

Parameter	Description	Details
btier # Below entry needs to be replaced with the appropriate cndbtier-namespace only if occndbtier_alertrules_dbtier_specific_<version>.yaml alert rules are patched. Otherwise leave it commented out. # - /etc/config/alerts<${CNDBTIER_NAMESPACE}>cndbtier - /etc/config/alertsscp - /etc/config/alertsnrf - /etc/config/alertspcf - /etc/		

Table 3-2 (Cont.) OSO Configuration Parameter

Parameter	Description	Details
config/alertsbsf - /etc/config/alertsudr - /etc/config/alertssepp - /etc/config/alertsnssf		

3.2.2 Alertmanager Configuration Parameters

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

Configure the `ocoso_csar_25_1_201_0_0_0_alm_custom_values.yaml` with the defined values.

> **Note**
>
> Configure the following parameters using Table 3-2 table:
>
> - `HELM-TEST-IMAGE-REPO-ADDRESS`
> - `USEASM`
> - `NAMESPACE`
> - `CONFIGMAP-RELOAD-REPO-ADDRESS`
> - `prefix`
> - `suffix`
> - `sidecar annotations`
> - `labels`

Table 3-3 OSO Alertmanager Configuration Parameters

Parameter	Description	Details
OSO-ALM-RELEASE-NAME-alm	This is a mandatory parameter. Indicates the Helm release name for installing the Alertmanager in OSO release. Example: oso-a	**Data Type:** String **Default Value:** NA **Range:** NA
ALERTMANAGER-REPO-ADDRESS	This is a mandatory parameter. Indicates the repo URL for alertmanager location. Example: Bastion:5000/occne.io/oso/alertmanager	**Data Type:** String **Default Value:** NA **Range:** NA
RETENTION-TIME-ALM	This is a mandatory parameter. Indicates the define retention time for alertmanager persistence. Example: 120h	**Data Type:** String **Default Value:** NA **Range:** NA
USE-DEFAULT-SVCACCOUNT-ALM	This is a mandatory parameter. Indicates if default svcaccount or own svcaccount is required for configuring OSO. If this value is true, default svcaccount at cluster scope is required. If this value is false, create your own svcaccount and configure OSO	**Data Type:** Boolean **Default Value:** true **Range:** true, false
ENABLE_DUAL_STACK	This is a conditional parameter. Indicates if dual stack support is available for alertmanager. If this value is true, dual stack support (IPV4 and IPV6) is enabled for alertmanager. If this value is false, dual stack support is not enabled for alertmanger.	**Data Type:** Boolean **Default Value:** true **Range:** true, false
ALM-STORAGE-CLASS	This is a conditional parameter. Indicates the alertmanager storage class configuration.	**Data Type:** String **Default Value:** standard **Range:** NA
ALM-PV-SIZE	This is a conditional parameter. Indicates the size for the alertmanager persistence volume.	**Data Type:** String **Default Value:** 2Gi **Range:** NA

Table 3-3 (Cont.) OSO Alertmanager Configuration Parameters

Parameter	Description	Details

`annotations: {}`	This is a conditional parameter.
	Specifies the custom annotations to be added to OSO deployments that are associated to disable the sidecar injections into OSO pods.
	Note: This parameter is mandatory if ASM is enabled.
	Example:
	`annotations:`
	`  - sidecar.istio.io/inject: "false"`
		Data Type: Boolean
		Default Value: false
		Range: true, false

3.2.3 Alert Config Configuration Parameters

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

Configure the `ocoso_csar_25_1_201_0_0_0_alr_config_custom_values.yaml` with the defined values.

Table 3-4 OSO AlertConfig Configuration Parameters

Parameter	Description	Details
`OSO-PROM-CONFIGMAP-NAME`	This is a mandatory parameter. Indicates the name of the OSO Prometheus ConfigMap. Example: oso-prom-svr	**Data Type:** String **Default Value:** NA **Range:** NA
`OSO_ALERT_CONFIG_IMAGE`	This is a mandatory parameter. Indicates the repo URL for the alert configuration location. Example: `occne.io/occne/oso_alert_config`	**Data Type:** String **Default Value:** NA **Range:** NA

`annotations: {}`	This is a conditional parameter.
	Specifies the custom annotations to be added to OSO deployments that are associated to disable the sidecar injections into OSO pods.
	Note: This parameter is mandatory if ASM is enabled.
	Example:
	`annotations:`
	`  - sidecar.istio.io/inject: "false"`
		Data Type: Boolean
		Default Value: false
		Range: true, false

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 Cloud Native Core, 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 25.1.1xx, 25.1.200 and 25.1.201, a small outage is expected.

5.1 Supported Upgrade Paths

The following table lists the supported upgrade path for OSO:

Table 5-1 Supported Upgrade Path

Source Release	Target Release
25.1.2xx	25.1.201
25.1.1xx	25.1.201

5.2 Prerequisites

Following are the prerequisites for upgrading OSO:

- Ensure that the system has OSO 25.1.1xx or 25.1.200 is 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_25_1_201_0_0_0_custom_values.yaml` file by performing the following steps. This .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_prom_25_1_201_0_0_0_custom_values.yaml` and `ocoso_csar_alm_25_1_201_0_0_0_custom_values.yaml` files 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 <OSO release-name for prometheus> <path to the new prom-OSO
helm chart> -n <oso-namespace> -f <path to
ocoso_csar_prom_25.1.201_custom_values.yaml file>
$ helm upgrade <OSO release-name for alertmanager> <path to the new alm-
OSO helm chart> -n <oso-namespace> -f <path to
ocoso_csar_alm_25.1.201_custom_values.yaml file>
```

> **Note**
>
> To install the alert config feature, see Automated Configuration of NF Alerts.

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 Cloud Native Core, Operations Services Overlay (OSO) deployment to previous releases.

OSO does not support rollback from 25.1.201 to previous releases.

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:

   ```
   $ helm 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
```

B
Prometheus Vertical Scaling

This section describes the procedure for vertical scaling of Prometheus.

To scale Prometheus deployments, follow these steps:

1. Get the list of deployments and identify the OSO Prometheus deployment with the suffix `prom-svr`:

   ```
   # To list all the deployments in the OSO namespace
   $ kubectl -n <OSO_namespace> get deployments
   # To filter the deployment name by its suffix
   $ kubectl -n <OSO_namespace> get deployments | grep prom-svr
   ```

2. Edit the OSO deployment using the following command:

   > **Note**
   >
   > This will open a *vi* editor with the deployment's yaml definition.

   ```
   $ kubectl -n <OSO_namespace> edit deployment <oso_deployment_name>-prom-svr
   ```

3. Find the `resources` section for the `prom-svr` container in the edit mode of deployment, and edit the amount of resources as per the requirements.

   ```
   name: prom-svr
   ports:
   ... # ports definitions
   readinessProbe:
   ... # readiness probe definition
   resources:
     limits:
       cpu: "2"
       memory: 4Gi
     requests:
       cpu: "2"
       memory: 4Gi
   ```

4. Save and quit from the editor after making the required changes in the yaml file for the CPU and memory. In case of any errors while editing, the editor opens again and error message appears at the top of the yaml file as a comment.

   > **Note**
   >
   > If any of these objects have two containers each, you will find two resources sections. For more information about how to assign resources, see https://kubernetes.io/docs/concepts/configuration/manage-resources-containers/.

C

Creating Backup of Prometheus Time Series Database (TSDB) Using Snapshot Utility

This section details how users can create backup of Prometheus Time Series Database (TSDB) using the snapshot utility.

Capturing TSDB Snapshots

Perform the following steps to capture the TSDB snapshots:

1. Enable the `web.enable-admin-api` flag provided in `ocoso_csar_25_1_201_0_0_0_prom_custom_values.yaml`:

   ```
   extraFlags:
             - web.enable-lifecycle
             ## web.enable-admin.api flag controls access to the
   administrative HTTP API which includes functionality such as
             ## deleting time series. This is disabled by default.
             - web.enable-admin-api
   ```

2. Install OSO using Installing OSO Using CSAR.
3. Use the ephemeral (Debug) container to capture the snapshot and export it out of Prometheus. After the export is done successfully, ephemeral container will be exited and stopped.

   > **Note**
   >
   > - Take the backup of current Prometheus data. You can wait for a couple days for Prometheus database to fill up with the required data or until OSO's retention period is over (the default period is 7 days). Then, perform the following steps to take the backup.
   > - Use the `occne.io/occne/oso_snapshot:25.1.201` image for ephemeral container. This image is created for the purpose of handling the creation and removal of the TSDB snapshots. You must load and push this image into the customer's central or system registry.

4. Capture snapshots by connecting the Debug container to the Prometheus server. When OSO is configured with `CLUSTER_NAME_PREFIX`, run the following command:

   ```
   $ kubectl -n <oso-namespace> debug <oso-prom-pod-name> -it --
   image=<oso_snapshot_image_url> --target=<name of Prometheus server
   container> --env OSO_PROMETHEUS_SERVICE_NAME=<oso-prometheus-service-name>
   --env CLUSTER_NAME_PREFIX=<cluster-name-prefix> &
   ```

When OSO is **not** configured with `CLUSTER_NAME_PREFIX`, provide only the `OSO_PROMETHEUS_SERVICE_NAME` and run the following command:

```
$ kubectl -n <oso-namespace> debug <oso-prom-pod-name> -it --
image=<oso_snapshot_image_url> --target=<name of Prometheus server
container> --env OSO_PROMETHEUS_SERVICE_NAME=<oso-prometheus-service-name>
&
```

For example:

```
$ kubectl -n oso debug oso-p-prom-svr-55f8d47c74-4vwfx -it --image=occne-
repo-host:5000/occne.io/occne/oso_snapshot:25_1_201 --target=prom-svr --
env OSO_PROMETHEUS_SERVICE_NAME=oso-p-prom-svr --env
CLUSTER_NAME_PREFIX=occne3-n2 &

Targeting container "prom-svr". If you don't see processes from this
container it may be because the container runtime doesn't support this
feature.
--profile=legacy is deprecated and will be removed in the future. It is
recommended to explicitly specify a profile, for example "--
profile=general".
Defaulting debug container name to debugger-x25bq.
If you don't see a command prompt, try pressing enter.

[1]+  Stopped        kubectl -n oso debug oso-p-prom-svr-55f8d47c74-4vwfx -
it --image=occne-repo-host:5000/occne.io/occne/oso_snapshot:25_1_201 --
target=prom-svr --env OSO_PROMETHEUS_SERVICE_NAME=oso-p-prom-svr --env
CLUSTER_NAME_PREFIX=occne3-n2
```

> **Note**
>
> There is an ampersand (&) character at the end of the command. This character indicates that the process is running in the background and allows the terminal available for other processes. In the above example, "`Defaulting debug container name to debugger-x25bq.`", the name "`debugger-x25bq`" indicates the ephemeral container name of the active container. This is used to copy the snapshot out of the pod and to remove it later.

5. Run the below command to view the logs, when the snapshot process is running in the background. The output should look like below with "`200 OK`" code.

```
kubectl -n <oso-namespace> logs <oso-prom-pod-name> -c <ephemeral-
container-name>
```

For example:

```
$ kubectl -n oso logs oso-p-prom-svr-55f8d47c74-4vwfx -c debugger-x25bq
```

Sample output:

```
total 36K
drwxr-sr-x. 2 nobody nobody 4.0K Aug 19 17:12 chunks_head
```

```
-rw-r--r--. 1 nobody nobody    0 Aug 19 17:12 lock
drwxrws---. 2 root   nobody  16K Aug 19 17:12 lost+found
-rw-r--r--. 1 nobody nobody  20K Aug 19 17:12 queries.active
-rw-r--r--. 1 nobody nobody 7.2K Aug 20 16:53 snapshots.log
drwxr-sr-x. 2 nobody nobody 4.0K Aug 19 17:12 wal
  % Total    % Received % Xferd  Average Speed   Time    Time     Time
Current
                                 Dload  Upload   Total   Spent    Left
Speed
  0     0    0     0    0     0      0      0 --:--:-- --:--:--
--:--:--     0*   Trying 10.233.11.145:80...
* Connected to oso-p-prom-svr (10.233.11.145) port 80 (#0)
> POST /occne3-n2/prometheus/api/v1/admin/tsdb/snapshot HTTP/1.1
> Host: oso-p-prom-svr
> User-Agent: curl/7.76.1
> Accept: */*
>
* Mark bundle as not supporting multiuse
< HTTP/1.1 200 OK
< Content-Type: application/json
< Date: Wed, 20 Aug 2025 16:53:24 GMT
< Content-Length: 72
<
{ [72 bytes data]
100    72  100    72    0     0   1440      0 --:--:-- --:--:-- --:--:--
1440
* Connection #0 to host oso-p-prom-svr left intact
```

6. Verify if the log of the curl command that issues the snapshot creation logic inside debug container is successful, by looking for a "`< HTTP/1.1 200 OK`" in the above sample output. Export the tarball (.tgz artifact) out of the debug container using the following command.

   ```
   $ kubectl cp <oso-namespace>/<oso-prom-svr-pod-name>:/proc/1/root/data/
   snapshots.tgz -c <debug-container-name> /tmp/<snapshot-folder-name>/
   snapshots.tgz
   ```

Figure C-1 Exporting tgz artifact from Debug container

7. Get the snapshot process back to foreground, once the snapshots tar is available in the local system, by running the following command:

   ```
   $ fg $(jobs | awk -F '[][]' '/oso_snapshot/{print $2}')
   ```

   For example:

   ```
   $ fg $(jobs | awk -F '[][]' '/oso_snapshot/{print $2}')
   ```

Sample output:

```
$ fg $(jobs | awk -F '[][]' '/oso_snapshot/{print $2}')
kubectl -n oso debug oso-p-prom-svr-55f8d47c74-4vwfx -it --image=occne-
repo-host:5000/occne.io/occne/oso_snapshot:25_1_201 --target=prom-svr --
env OSO_PROMETHEUS_SERVICE_NAME=oso-p-prom-svr --env
CLUSTER_NAME_PREFIX=occne3-n2
```

The above step leaves the terminal in a waiting state.

Press **Enter** in the terminal to finalize the process and to see the full log of the snapshot creation process.

```
------------------------------ Snapshot procedure started
---------------------------------

STEP 1: CHECKING FOR EXISTING SNAPSHOT ARCHIVES AND CLEANING UP IF
NECESSARY

Listing current contents of the directory for snapshot archives:

total 32K
drwxrws---. 2 root   nobody  16K Aug 19 17:12 lost+found
-rw-r--r--. 1 nobody nobody  20K Aug 19 17:12 queries.active
drwxr-sr-x. 2 nobody nobody 4.0K Aug 19 17:12 wal
-rw-r--r--. 1 nobody nobody    0 Aug 19 17:12 lock
drwxr-sr-x. 2 nobody nobody 4.0K Aug 19 17:12 chunks_head
-rw-r--r--. 1 nobody nobody  239 Aug 20 16:51 snapshots.log

No previous snapshots.tgz found. All clear!

No previous snapshots directory found. All clear!

STEP 2: CAPTURING SNAPSHOT OF CURRENT PROMETHEUS DB USING ADMINISTRATIVE
API

Wed Aug 20 16:51:01 UTC 2025

Sending POST request to Prometheus API to create a snapshot with Cluster
Name Prefix...

Executing: curl -vvv -XPOST "http://oso-p-prom-svr/occne3-n2/
prometheus/api/v1/admin/tsdb/snapshot"

  % Total    % Received % Xferd  Average Speed   Time    Time     Time
Current
                                 Dload  Upload   Total   Spent    Left
Speed
  0     0    0     0    0     0      0      0 --:--:-- --:--:--
--:--:--     0*   Trying 10.233.11.145:80...
* Connected to oso-p-prom-svr (10.233.11.145) port 80 (#0)
> POST /occne3-n2/prometheus/api/v1/admin/tsdb/snapshot HTTP/1.1
> Host: oso-p-prom-svr
> User-Agent: curl/7.76.1
> Accept: */*
```

```
>
* Mark bundle as not supporting multiuse
< HTTP/1.1 200 OK
< Content-Type: application/json
< Date: Wed, 20 Aug 2025 16:51:01 GMT
< Content-Length: 72
<
{ [72 bytes data]
100    72  100    72    0     0    911      0 --:--:-- --:--:-- --:--:--
911
* Connection #0 to host oso-p-prom-svr left intact
{"status":"success","data":\{"name":"20250820T165101Z-0eb533ff8d43ed64"}}

 Snapshot is successfully created!

 Directory contents of Prometheus DB AFTER successful Snapshot:

total 36K
drwxrws---. 2 root   nobody  16K Aug 19 17:12 lost+found
-rw-r--r--. 1 nobody nobody  20K Aug 19 17:12 queries.active
drwxr-sr-x. 2 nobody nobody 4.0K Aug 19 17:12 wal
-rw-r--r--. 1 nobody nobody    0 Aug 19 17:12 lock
drwxr-sr-x. 2 nobody nobody 4.0K Aug 19 17:12 chunks_head
drwxr-sr-x. 3 nobody nobody 4.0K Aug 20 16:51 snapshots
-rw-r--r--. 1 nobody nobody 2.0K Aug 20 16:51 snapshots.log

STEP 3: CREATING EXPORTABLE ARCHIVE OF THE SNAPSHOT DATA

 Snapshots is successfully archived!

total 40K
drwxrws---. 2 root   nobody  16K Aug 19 17:12 lost+found
-rw-r--r--. 1 nobody nobody  20K Aug 19 17:12 queries.active
drwxr-sr-x. 2 nobody nobody 4.0K Aug 19 17:12 wal
-rw-r--r--. 1 nobody nobody    0 Aug 19 17:12 lock
drwxr-sr-x. 2 nobody nobody 4.0K Aug 19 17:12 chunks_head
drwxr-sr-x. 3 nobody nobody 4.0K Aug 20 16:51 snapshots
-rw-r--r--. 1 nobody nobody  162 Aug 20 16:51 snapshots.tgz
-rw-r--r--. 1 nobody nobody 2.5K Aug 20 16:51 snapshots.log
Non-interactive shell detected. Skipping interactive prompt...

---------------------------------- Snapshot creation procedure completed
--------------------------------------

------------------------------- Snapshot procedure started
----------------------------------

STEP 1: CHECKING FOR EXISTING SNAPSHOT ARCHIVES AND CLEANING UP IF
NECESSARY

Listing current contents of the directory for snapshot archives:

total 40K
drwxrws---. 2 root   nobody  16K Aug 19 17:12 lost+found
-rw-r--r--. 1 nobody nobody  20K Aug 19 17:12 queries.active
drwxr-sr-x. 2 nobody nobody 4.0K Aug 19 17:12 wal
```

```
-rw-r--r--. 1 nobody nobody    0 Aug 19 17:12 lock
drwxr-sr-x. 2 nobody nobody 4.0K Aug 19 17:12 chunks_head
drwxr-sr-x. 3 nobody nobody 4.0K Aug 20 16:51 snapshots
-rw-r--r--. 1 nobody nobody  162 Aug 20 16:51 snapshots.tgz
-rw-r--r--. 1 nobody nobody 3.3K Aug 20 16:53 snapshots.log

Found snapshots.tgz archive. Deleting...

 Successfully deleted!

Found snapshots directory. Deleting...

 Successfully deleted!

STEP 2: CAPTURING SNAPSHOT OF CURRENT PROMETHEUS DB USING ADMINISTRATIVE
API

Wed Aug 20 16:53:01 UTC 2025

Sending POST request to Prometheus API to create a snapshot with Cluster
Name Prefix...

Executing: curl -vvv -XPOST "http://oso-p-prom-svr/occne3-n2/
prometheus/api/v1/admin/tsdb/snapshot"

  % Total    % Received % Xferd  Average Speed   Time    Time     Time
Current
                                 Dload  Upload   Total   Spent    Left
Speed
  0     0    0     0    0     0      0      0 --:--:-- --:--:--
--:--:--     0*   Trying 10.233.11.145:80...
* Connected to oso-p-prom-svr (10.233.11.145) port 80 (#0)
> POST /occne3-n2/prometheus/api/v1/admin/tsdb/snapshot HTTP/1.1
> Host: oso-p-prom-svr
> User-Agent: curl/7.76.1
> Accept: */*
>
* Mark bundle as not supporting multiuse
< HTTP/1.1 200 OK
< Content-Type: application/json
< Date: Wed, 20 Aug 2025 16:53:01 GMT
< Content-Length: 72
<
{ [72 bytes data]
100    72  100    72    0     0   2000      0 --:--:-- --:--:-- --:--:--
1945
* Connection #0 to host oso-p-prom-svr left intact
{"status":"success","data":\{"name":"20250820T165301Z-0572f8dd893c2028"}}

 Snapshot is successfully created!

 Directory contents of Prometheus DB AFTER successful Snapshot:

total 40K
drwxrws---. 2 root   nobody  16K Aug 19 17:12 lost+found
-rw-r--r--. 1 nobody nobody  20K Aug 19 17:12 queries.active
```

```
drwxr-sr-x. 2 nobody nobody 4.0K Aug 19 17:12 wal
-rw-r--r--. 1 nobody nobody    0 Aug 19 17:12 lock
drwxr-sr-x. 2 nobody nobody 4.0K Aug 19 17:12 chunks_head
drwxr-sr-x. 3 nobody nobody 4.0K Aug 20 16:53 snapshots
-rw-r--r--. 1 nobody nobody 5.3K Aug 20 16:53 snapshots.log

STEP 3: CREATING EXPORTABLE ARCHIVE OF THE SNAPSHOT DATA

 Snapshots is successfully archived!

total 44K
drwxrws---. 2 root   nobody  16K Aug 19 17:12 lost+found
-rw-r--r--. 1 nobody nobody  20K Aug 19 17:12 queries.active
drwxr-sr-x. 2 nobody nobody 4.0K Aug 19 17:12 wal
-rw-r--r--. 1 nobody nobody    0 Aug 19 17:12 lock
drwxr-sr-x. 2 nobody nobody 4.0K Aug 19 17:12 chunks_head
drwxr-sr-x. 3 nobody nobody 4.0K Aug 20 16:53 snapshots
-rw-r--r--. 1 nobody nobody  161 Aug 20 16:53 snapshots.tgz
-rw-r--r--. 1 nobody nobody 5.8K Aug 20 16:53 snapshots.log
Non-interactive shell detected. Skipping interactive prompt...

---------------------------------- Snapshot creation procedure completed
----------------------------------------

------------------------------ Snapshot procedure started
---------------------------------

STEP 1: CHECKING FOR EXISTING SNAPSHOT ARCHIVES AND CLEANING UP IF
NECESSARY

Listing current contents of the directory for snapshot archives:

total 44K
drwxrws---. 2 root   nobody  16K Aug 19 17:12 lost+found
-rw-r--r--. 1 nobody nobody  20K Aug 19 17:12 queries.active
drwxr-sr-x. 2 nobody nobody 4.0K Aug 19 17:12 wal
-rw-r--r--. 1 nobody nobody    0 Aug 19 17:12 lock
drwxr-sr-x. 2 nobody nobody 4.0K Aug 19 17:12 chunks_head
drwxr-sr-x. 3 nobody nobody 4.0K Aug 20 16:53 snapshots
-rw-r--r--. 1 nobody nobody  161 Aug 20 16:53 snapshots.tgz
-rw-r--r--. 1 nobody nobody 6.6K Aug 20 16:53 snapshots.log

Found snapshots.tgz archive. Deleting...

 Successfully deleted!

Found snapshots directory. Deleting...

 Successfully deleted!

STEP 2: CAPTURING SNAPSHOT OF CURRENT PROMETHEUS DB USING ADMINISTRATIVE
API

Wed Aug 20 16:53:24 UTC 2025

Sending POST request to Prometheus API to create a snapshot with Cluster
```

```
Name Prefix...

Executing: curl -vvv -XPOST "http://oso-p-prom-svr/occne3-n2/
prometheus/api/v1/admin/tsdb/snapshot"

  % Total    % Received % Xferd  Average Speed   Time    Time     Time
Current
                                 Dload  Upload   Total   Spent    Left
Speed
  0     0    0     0    0     0      0      0 --:--:-- --:--:--
--:--:--     0*   Trying 10.233.11.145:80...
* Connected to oso-p-prom-svr (10.233.11.145) port 80 (#0)
> POST /occne3-n2/prometheus/api/v1/admin/tsdb/snapshot HTTP/1.1
> Host: oso-p-prom-svr
> User-Agent: curl/7.76.1
> Accept: */*
>
* Mark bundle as not supporting multiuse
< HTTP/1.1 200 OK
< Content-Type: application/json
< Date: Wed, 20 Aug 2025 16:53:24 GMT
< Content-Length: 72
<
{ [72 bytes data]
100    72  100    72    0     0   1440      0 --:--:-- --:--:-- --:--:--
1440
* Connection #0 to host oso-p-prom-svr left intact
{"status":"success","data":\{"name":"20250820T165324Z-3e8dc6572924e398"}}

 Snapshot is successfully created!

 Directory contents of Prometheus DB AFTER successful Snapshot:

total 44K
drwxrws---. 2 root   nobody  16K Aug 19 17:12 lost+found
-rw-r--r--. 1 nobody nobody  20K Aug 19 17:12 queries.active
drwxr-sr-x. 2 nobody nobody 4.0K Aug 19 17:12 wal
-rw-r--r--. 1 nobody nobody    0 Aug 19 17:12 lock
drwxr-sr-x. 2 nobody nobody 4.0K Aug 19 17:12 chunks_head
drwxr-sr-x. 3 nobody nobody 4.0K Aug 20 16:53 snapshots
-rw-r--r--. 1 nobody nobody 8.5K Aug 20 16:53 snapshots.log

STEP 3: CREATING EXPORTABLE ARCHIVE OF THE SNAPSHOT DATA

 Snapshots is successfully archived!

total 48K
drwxrws---. 2 root   nobody  16K Aug 19 17:12 lost+found
-rw-r--r--. 1 nobody nobody  20K Aug 19 17:12 queries.active
drwxr-sr-x. 2 nobody nobody 4.0K Aug 19 17:12 wal
-rw-r--r--. 1 nobody nobody    0 Aug 19 17:12 lock
drwxr-sr-x. 2 nobody nobody 4.0K Aug 19 17:12 chunks_head
drwxr-sr-x. 3 nobody nobody 4.0K Aug 20 16:53 snapshots
-rw-r--r--. 1 nobody nobody  163 Aug 20 16:53 snapshots.tgz
-rw-r--r--. 1 nobody nobody 9.0K Aug 20 16:53 snapshots.log
```

```
---------------------------------- Snapshot creation procedure completed
---------------------------------------

Session ended, the ephemeral container will not be restarted but may be
reattached using 'kubectl attach oso-p-prom-svr-55f8d47c74-4vwfx -c
debugger-x25bq -i -t' if it is still running
```

8. (Mandatory) Clean up the snapshot archive. Run the following command to remove the snapshots created.

> **Warning**
>
> Failing to perform the following step will leave a snapshot hanging in your system and may fill up your system's storage.

```
kubectl -n <oso-namespace> debug <oso-prom-pod-name> -it --
image=<oso_snapshot_image_url> --target=prom-svr --env REMOVE=yes
```

9. Verify the output of the above command to confirm if the snapshots were removed. For example:

```
$ kubectl -n oso debug oso-p-prom-svr-55f8d47c74-4vwfx -it --image=occne-
repo-host:5000/occne.io/occne/oso_snapshot:25_1_201 --target=prom-svr --
env REMOVE=yes
```

Sample output:

```
Targeting container "prom-svr". If you don't see processes from this
container it may be because the container runtime doesn't support this
feature.
--profile=legacy is deprecated and will be removed in the future. It is
recommended to explicitly specify a profile, for example "--
profile=general".
Defaulting debug container name to debugger-fc5q6.

---------------------------------- REMOVE flag is set to 'yes'. Cleaning
up snapshot archives -----------------------------------

 Prometheus DB Directory contents BEFORE cleanup:
total 48K
drwxr-sr-x. 2 nobody nobody 4.0K Aug 19 17:12 chunks_head
-rw-r--r--. 1 nobody nobody    0 Aug 19 17:12 lock
drwxrws---. 2 root   nobody  16K Aug 19 17:12 lost+found
-rw-r--r--. 1 nobody nobody  20K Aug 19 17:12 queries.active
drwxr-sr-x. 3 nobody nobody 4.0K Aug 20 16:53 snapshots
-rw-r--r--. 1 nobody nobody 9.5K Aug 20 16:54 snapshots.log
-rw-r--r--. 1 nobody nobody  163 Aug 20 16:53 snapshots.tgz
drwxr-sr-x. 2 nobody nobody 4.0K Aug 19 17:12 wal

 Snapshots archives removed successfully!
```

```
 Prometheus DB Directory contents AFTER cleanup:
total 28K
drwxr-sr-x. 2 nobody nobody 4.0K Aug 19 17:12 chunks_head
-rw-r--r--. 1 nobody nobody    0 Aug 19 17:12 lock
drwxrws---. 2 root   nobody  16K Aug 19 17:12 lost+found
-rw-r--r--. 1 nobody nobody  20K Aug 19 17:12 queries.active
drwxr-sr-x. 2 nobody nobody 4.0K Aug 19 17:12 wal

---------------------------------- Snapshot cleanup process completed
-------------------------------------
```

10. (Mandatory)Validate the contents of the snapshot by running the following command on the snapshot file:

    ```
    $ tar tvf snapshots.tgz
    ```

    For example:

    ```
    $ tar tvf snapshots.tgz
    drwxr-sr-x nobody/nobody      0 2025-08-25 21:11 snapshots/
    drwxr-sr-x nobody/nobody      0 2025-08-25 21:11 snapshots/
    20250825T211118Z-0aa128d1c1f37e71/
    drwxr-sr-x nobody/nobody      0 2025-08-25 21:11 snapshots/
    20250107T084350Z-2a422bf3ea6cee2c/01JGZYYEXF1DR08XFAHX99Z1EW/
    drwxr-sr-x nobody/nobody      0 2025-08-25 21:11 snapshots/
    20250107T084350Z-2a422bf3ea6cee2c/01JGZYYEXF1DR08XFAHX99Z1EW/chunks/
    -rw-r--r-- nobody/nobody  14004 2025-08-25 21:11 snapshots/
    20250107T084350Z-2a422bf3ea6cee2c/01JGZYYEXF1DR08XFAHX99Z1EW/chunks/000001
    -rw-r--r-- nobody/nobody      9 2025-08-25 21:11 snapshots/
    20250107T084350Z-2a422bf3ea6cee2c/01JGZYYEXF1DR08XFAHX99Z1EW/tombstones
    -rw-r--r-- nobody/nobody  69072 2025-08-25 21:11 snapshots/
    20250107T084350Z-2a422bf3ea6cee2c/01JGZYYEXF1DR08XFAHX99Z1EW/index
    -rw-r--r-- nobody/nobody    273 2025-08-25 21:11 snapshots/
    20250107T084350Z-2a422bf3ea6cee2c/01JGZYYEXF1DR08XFAHX99Z1EW/meta.json
    ```

    > **Note**
    >
    > If the snapshot in your system does not have the structure shown above, delete the snapshot and try the process again.