



Red Hat OpenShift

Installing the Network Observability Operator

- [Network Observability without Loki](#)
- [Installing the Loki Operator](#)
 - [Creating a secret for Loki storage](#)
 - [Creating a LokiStack custom resource](#)
 - [Creating a new group for the cluster-admin user role](#)
 - [Custom admin group access](#)
 - [Loki deployment sizing](#)
 - [LokiStack ingestion limits and health alerts](#)
 - [Enabling multi-tenancy in Network Observability](#)
- [Installing the Network Observability Operator](#)
- [Important Flow Collector configuration considerations](#)
 - [Migrating removed stored versions of the FlowCollector CRD](#)
- [Installing Kafka \(optional\)](#)
- [Uninstalling the Network Observability Operator](#)

Installing Loki is a recommended prerequisite for using the Network Observability Operator. You can choose to use [Network Observability without Loki](#), but there are some considerations for doing this, described in the previously linked section.

The Loki Operator integrates a gateway that implements multi-tenancy and authentication with Loki for data flow storage. The `LokiStack` resource manages Loki, which is a scalable, highly-available, multi-tenant log aggregation system, and a web proxy with OpenShift Container Platform authentication. The `LokiStack` proxy uses OpenShift Container Platform authentication to enforce multi-tenancy and facilitate the saving and indexing of data in Loki log stores.



The Loki Operator can also be used for [configuring the LokiStack log store](#). The Network Observability Operator requires a dedicated LokiStack separate from the logging.

Network Observability without Loki

You can use Network Observability without Loki by not performing the Loki installation steps and skipping directly to "Installing the Network Observability Operator". If you only want to export flows to a Kafka consumer or IPFIX collector, or you only need dashboard metrics, then you do not need to install Loki or provide storage for Loki. The following table compares available features with and without Loki.

Table 1. Comparison of feature availability with and without Loki

	With Loki	Without Loki
Exporters	✓	✓
Multi-tenancy	✓	✗
Complete filtering and aggregations capabilities ^[1]	✓	✗
Partial filtering and aggregations capabilities ^[2]	✓	✓
Flow-based metrics and dashboards	✓	✓
Traffic flows view overview ^[3]	✓	✓
Traffic flows view table	✓	✗
Topology view	✓	✓
OpenShift Container Platform console Network Traffic tab integration	✓	✓

- Such as per pod.
- Such as per workload or namespace.
- Statistics on packet drops are only available with Loki.

Additional resources

- [Export enriched network flow data](#).

Installing the Loki Operator

The [Loki Operator versions 5.7+](#) are the supported Loki Operator versions for Network Observability; these versions provide the ability to create a `LokiStack` instance using the `openshift-network` tenant configuration mode and provide fully-automatic, in-cluster authentication and authorization support for Network Observability. There are several ways you can install Loki. One way is by using the OpenShift Container Platform web console Operator Hub.

Prerequisites

- Supported Log Store (AWS S3, Google Cloud Storage, Azure, Swift, Minio, OpenShift Data Foundation)
- OpenShift Container Platform 4.10+
- Linux Kernel 4.18+

Procedure

- In the OpenShift Container Platform web console, click **Operators** → **OperatorHub**.
- Choose **Loki Operator** from the list of available Operators, and click **Install**.
- Under **Installation Mode**, select **All namespaces on the cluster**.

Verification

- Verify that you installed the Loki Operator. Visit the **Operators** → **Installed Operators** page and look for **Loki Operator**.
- Verify that **Loki Operator** is listed with **Status** as **Succeeded** in all the projects.



To uninstall Loki, refer to the uninstallation process that corresponds with the method you used to install Loki. You might have remaining `ClusterRoles` and `ClusterRoleBindings`, data stored in object store, and persistent volume that must be removed.

Creating a secret for Loki storage

The Loki Operator supports a few log storage options, such as AWS S3, Google Cloud Storage, Azure, Swift, Minio, OpenShift Data Foundation. The following example shows how to create a secret for AWS S3 storage. The secret created in this example, `loki-s3`, is referenced in "Creating a LokiStack resource". You can create this secret in the web console or CLI.

- Using the web console, navigate to the **Project** → **All Projects** dropdown and select **Create Project**. Name the project `netobserv` and click **Create**.
- Navigate to the Import icon, **+**, in the top right corner. Paste your YAML file into the editor.

The following shows an example secret YAML file for S3 storage:

```
apiVersion: v1
kind: Secret
metadata:
  name: loki-s3
  namespace: netobserv    (1)
stringData:
  access_key_id: QUtJQUlPU0ZPRE5ON0VYQU1QTEUK
  access_key_secret:
d0phbHJYVXRuRkVNSS9LN01ERU5HL2JQeFJmaUNZRVhBTvBMRUtFWQo=
  bucketnames: s3-bucket-name
  endpoint: https://s3.eu-central-1.amazonaws.com
  region: eu-central-1
```

The installation examples in this documentation use the same namespace, **1 netobserv**, across all components. You can optionally use a different namespace for the different components

Verification

- Once you create the secret, you should see it listed under **Workloads** → **Secrets** in the web console.

Additional resources

- [Flow Collector API Reference](#)
- [Flow Collector sample resource](#)
- [Loki object storage](#)

Creating a LokiStack custom resource

You can deploy a `LokiStack` custom resource (CR) by using the web console or OpenShift CLI (`oc`) to create a namespace, or new project.

Procedure

- Navigate to **Operators** → **Installed Operators**, viewing **All projects** from the **Project** dropdown.
- Look for **Loki Operator**. In the details, under **Provided APIs**, select **LokiStack**.
- Click **Create LokiStack**.
- Ensure the following fields are specified in either **Form View** or **YAML view**:

```
apiVersion: loki.grafana.com/v1
kind: LokiStack
metadata:
  name: loki
  namespace: netobserv (1)
spec:
  size: 1x.small (2)
  storage:
    schemas:
      - version: v12
        effectiveDate: '2022-06-01'
    secret:
      name: loki-s3
      type: s3
  storageClassName: gp3 (3)
  tenants:
    mode: openshift-network
```

- The installation examples in this documentation use the same namespace,
- 1 `netobserv`, across all components. You can optionally use a different namespace.

Specify the deployment size. In the Loki Operator 5.8 and later versions, the supported size options for production instances of Loki are `1x.extra-small`, `1x.small`, or `1x.medium`.

2



It is not possible to change the number `1x` for the deployment size.

Use a storage class name that is available on the cluster for `ReadWriteOnce` access mode. You can use `oc get storageclasses` to see what is available on your cluster.

3



You must not reuse the same `LokiStack` CR that is used for logging.

- Click **Create**.

Creating a new group for the cluster-admin user role



Querying application logs for multiple namespaces as a `cluster-admin` user, where the sum total of characters of all of the namespaces in the cluster is greater than 5120, results in the error `Parse error: input size too long (XXXX > 5120)`. For better control over access to logs in `LokiStack`, make the `cluster-admin` user a member of the `cluster-admin` group. If the `cluster-admin` group does not exist, create it and add the desired users to it.

Use the following procedure to create a new group for users with `cluster-admin` permissions.

Procedure

- Enter the following command to create a new group:

```
$ oc adm groups new cluster-admin
```

- Enter the following command to add the desired user to the `cluster-admin` group:

```
$ oc adm groups add-users cluster-admin <username>
```

- Enter the following command to add `cluster-admin` user role to the group:

```
$ oc adm policy add-cluster-role-to-group cluster-admin
cluster-admin
```

Custom admin group access

If you need to see cluster-wide logs without necessarily being an administrator, or if you already have any group defined that you want to use here, you can specify a custom group using the `adminGroup` field. Users who are members of any group specified in the `adminGroups` field of the `LokiStack` custom resource (CR) have the same read access to logs as administrators.

Administrator users have access to all network logs across the cluster.

Example LokiStack CR

```
apiVersion: loki.grafana.com/v1
kind: LokiStack
metadata:
  name: loki
  namespace: netobserv
spec:
  tenants:
    mode: openshift-network (1)
    openshift:
      adminGroups: (2)
      - cluster-admin
      - custom-admin-group (3)
```

- 1 Custom admin groups are only available in this mode.
- 2 Entering an empty list `[]` value for this field disables admin groups.
- 3 Overrides the default groups (`system:cluster-admins`, `cluster-admin`, `dedicated-admin`)

Loki deployment sizing

Sizing for Loki follows the format of `1x.<size>` where the value `1x` is number of instances and `<size>` specifies performance capabilities.



It is not possible to change the number 1x for the deployment size.

Table 2. Loki sizing

	1x.demo	1x.extra-small	1x.small	1x.medium
Data transfer	Demo use only	100GB/day	500GB/day	2TB/day
Queries per second (QPS)	Demo use only	1-25 QPS at 200ms	25-50 QPS at 200ms	25-75 QPS at 200ms
Replication factor	None	2	2	2
Total CPU requests	None	14 vCPUs	34 vCPUs	54 vCPUs
Total memory requests	None	31Gi	67Gi	139Gi
Total disk requests	40Gi	430Gi	430Gi	590Gi

LokiStack ingestion limits and health alerts

The LokiStack instance comes with default settings according to the configured size. It is possible to override some of these settings, such as the ingestion and query limits. You might want to update them if you get Loki errors showing up in the Console plugin, or in `flowlogs-pipeline` logs. An automatic alert in the web console notifies you when these limits are reached.

Here is an example of configured limits:


```
spec:
  limits:
    global:
      ingestion:
        ingestionBurstSize: 40
        ingestionRate: 20
        maxGlobalStreamsPerTenant: 25000
      queries:
        maxChunksPerQuery: 2000000
        maxEntriesLimitPerQuery: 10000
        maxQuerySeries: 3000
```

For more information about these settings, see the [LokiStack API reference](#).

Enabling multi-tenancy in Network Observability

Multi-tenancy in the Network Observability Operator allows and restricts individual user access, or group access, to the flows stored in Loki. Access is enabled for project admins. Project admins who have limited access to some namespaces can access flows for only those namespaces.

Prerequisite

- You have installed at least [Loki Operator version 5.7](#)
- You must be logged in as a project administrator

Procedure

- Authorize reading permission to `user1` by running the following command:

```
$ oc adm policy add-cluster-role-to-user netobserv-reader user1
```

Now, the data is restricted to only allowed user namespaces. For example, a user that has access to a single namespace can see all the flows internal to this namespace, as well as flows going from and to this namespace. Project admins have access to the Administrator perspective in the OpenShift Container Platform console to access the Network Flows Traffic page.

Installing the Network Observability Operator

You can install the Network Observability Operator using the OpenShift Container Platform web console Operator Hub. When you install the Operator, it provides the `FlowCollector` custom resource definition (CRD). You can set specifications in the web console when you create the `FlowCollector`.



The actual memory consumption of the Operator depends on your cluster size and the number of resources deployed. Memory consumption might need to be adjusted accordingly. For more information refer to "Network Observability controller manager pod runs out of memory" in the "Important Flow Collector configuration considerations" section.

Prerequisites

- If you choose to use Loki, install the [Loki Operator version 5.7+](#).
- You must have `cluster-admin` privileges.
- One of the following supported architectures is required: `amd64`, `ppc64le`, `arm64`, or `s390x`.
- Any CPU supported by Red Hat Enterprise Linux (RHEL) 9.
- Must be configured with OVN-Kubernetes or OpenShift SDN as the main network plugin, and optionally using secondary interfaces with Multus and SR-IOV.



Additionally, this installation example uses the `netobserv` namespace, which is used across all components. You can optionally use a different namespace.

Procedure

- In the OpenShift Container Platform web console, click **Operators** → **OperatorHub**.
- Choose **Network Observability Operator** from the list of available Operators in the **OperatorHub**, and click **Install**.

- Select the checkbox **Enable Operator recommended cluster monitoring** on this Namespace.
- Navigate to **Operators** → **Installed Operators**. Under Provided APIs for Network Observability, select the **Flow Collector** link.
- Navigate to the **Flow Collector** tab, and click **Create FlowCollector**. Make the following selections in the form view:
 - **spec.agent.ebpf.Sampling**: Specify a sampling size for flows. Lower sampling sizes will have higher impact on resource utilization. For more information, see the "FlowCollector API reference", `spec.agent.ebpf`.
 - If you are using Loki, set the following specifications:
 - **spec.loki.mode**: Set this to the `LokiStack` mode, which automatically sets URLs, TLS, cluster roles and a cluster role binding, as well as the `authToken` value. Alternatively, the `Manual` mode allows more control over configuration of these settings.
 - **spec.loki.lokiStack.name**: Set this to the name of your `LokiStack` resource. In this documentation, `loki` is used.
 - Optional: If you are in a large-scale environment, consider configuring the `FlowCollector` with Kafka for forwarding data in a more resilient, scalable way. See "Configuring the Flow Collector resource with Kafka storage" in the "Important Flow Collector configuration considerations" section.
 - Optional: Configure other optional settings before the next step of creating the `FlowCollector`. For example, if you choose not to use Loki, then you can configure exporting flows to Kafka or IPFIX. See "Export enriched network flow data to Kafka and IPFIX" and more in the "Important Flow Collector configuration considerations" section.
- Click **Create**.

Verification

To confirm this was successful, when you navigate to **Observe** you should see **Network Traffic** listed in the options.

In the absence of **Application Traffic** within the OpenShift Container Platform cluster, default filters might show that there are "No results", which results in no visual flow. Beside the filter selections, select **Clear all filters** to see the flow.

Important Flow Collector configuration considerations

Once you create the `FlowCollector` instance, you can reconfigure it, but the pods are terminated and recreated again, which can be disruptive. Therefore, you can consider configuring the following options when creating the `FlowCollector` for the first time:

- [Configuring the Flow Collector resource with Kafka](#)
- [Export enriched network flow data to Kafka or IPFIX](#)
- [Configuring monitoring for SR-IOV interface traffic](#)
- [Working with conversation tracking](#)
- [Working with DNS tracking](#)
- [Working with packet drops](#)

Additional resources

For more general information about Flow Collector specifications and the Network Observability Operator architecture and resource use, see the following resources:

- [Flow Collector API Reference](#)
- [Flow Collector sample resource](#)
- [Resource considerations](#)
- [Troubleshooting Network Observability controller manager pod runs out of memory](#)
- [Network Observability architecture](#)

Migrating removed stored versions of the FlowCollector CRD

Network Observability Operator version 1.6 removes the old and deprecated `v1alpha1` version of the `FlowCollector` API. If you previously installed this version on your cluster, it might still be referenced in the `storedVersion` of the `FlowCollector` CRD, even if it is removed from the etcd store, which blocks the upgrade process. These references need to be manually removed.

There are two options to remove stored versions:

- Use the Storage Version Migrator Operator.
- Uninstall and reinstall the Network Observability Operator, ensuring that the installation is in a clean state.

Prerequisites

- You have an older version of the Operator installed, and you want to prepare your cluster to install the latest version of the Operator. Or you have attempted to install the Network Observability Operator 1.6 and run into the error: `Failed risk of data loss updating "flowcollectors.flows.netobserv.io": new CRD removes version v1alpha1 that is listed as a stored version on the existing CRD.`

Procedure

- Verify that the old `FlowCollector` CRD version is still referenced in the `storedVersion`:

```
$ oc get crd flowcollectors.flows.netobserv.io -  
ojsonpath='{.status.storedVersions}'
```

- If `v1alpha1` appears in the list of results, proceed with **Step a** to use the Kubernetes Storage Version Migrator or **Step b** to uninstall and reinstall the CRD and the Operator.
 - **Option 1: Kubernetes Storage Version Migrator**: Create a YAML to define the `StorageVersionMigration` object, for example `migrate-flowcollector-v1alpha1.yaml`:

```
apiVersion: migration.k8s.io/v1alpha1
kind: StorageVersionMigration
metadata:
  name: migrate-flowcollector-v1alpha1
spec:
  resource:
    group: flows.netobserv.io
    resource: flowcollectors
    version: v1alpha1
```

- Save the file.
- Apply the `StorageVersionMigration` by running the following command:

```
$ oc apply -f migrate-flowcollector-v1alpha1.yaml
```

- Update the `FlowCollector` CRD to manually remove `v1alpha1` from the `storedVersion`:

```
$ oc edit crd flowcollectors.flows.netobserv.io
```

- **Option 2: Reinstall:** Save the Network Observability Operator 1.5 version of the `FlowCollector` CR to a file, for example `flowcollector-1.5.yaml`.

```
$ oc get flowcollector cluster -o yaml > flowcollector-1.5.yaml
```

- Follow the steps in "Uninstalling the Network Observability Operator", which uninstalls the Operator and removes the existing `FlowCollector` CRD.
- Install the Network Observability Operator latest version, 1.6.0.
- Create the `FlowCollector` using backup that was saved in Step b.

Verification

- Run the following command:

```
$ oc get crd flowcollectors.flows.netobserv.io -ojsonpath='{.status.storedVersions}'
```

The list of results should no longer show `v1alpha1` and only show the latest version, `v1beta1`.

Additional resources

- [Kubernetes Storage Version Migrator Operator](#)

Installing Kafka (optional)

The Kafka Operator is supported for large scale environments. Kafka provides high-throughput and low-latency data feeds for forwarding network flow data in a more resilient, scalable way. You can install the Kafka Operator as [Red Hat AMQ Streams](#) from the Operator Hub, just as the Loki Operator and Network Observability Operator were installed. Refer to "Configuring the FlowCollector resource with Kafka" to configure Kafka as a storage option.



To uninstall Kafka, refer to the uninstallation process that corresponds with the method you used to install.


Additional resources



[Configuring the FlowCollector resource with Kafka.](#)

Uninstalling the Network Observability Operator

You can uninstall the Network Observability Operator using the OpenShift Container Platform web console Operator Hub, working in the **Operators** → **Installed Operators** area.

Procedure

- Remove the `FlowCollector` custom resource.
 - Click **Flow Collector**, which is next to the **Network Observability Operator** in the **Provided APIs** column.
 - Click the options menu  for the **cluster** and select **Delete FlowCollector**.
- Uninstall the Network Observability Operator.
 - Navigate back to the **Operators** → **Installed Operators** area.

- Click the options menu  next to the **Network Observability Operator** and select **Uninstall Operator**.
- **Home** → **Projects** and select openshift-netobserv-operator
- Navigate to **Actions** and select **Delete Project**
- Remove the **FlowCollector** custom resource definition (CRD).
 - Navigate to **Administration** → **CustomResourceDefinitions**.
 - Look for **FlowCollector** and click the options menu .
 - Select **Delete CustomResourceDefinition**.



The Loki Operator and Kafka remain if they were installed and must be removed separately. Additionally, you might have remaining data stored in an object store, and a persistent volume that must be removed.