OpenShift Dedicated 4

Networking

Configuring OpenShift Dedicated networking

Last Updated: 2022-08-03

OpenShift Dedicated 4 Networking

Configuring OpenShift Dedicated networking

Legal Notice

Abstract

This document provides information about networking for OpenShift Dedicated clusters.

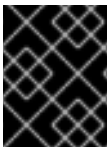
Table of Contents

CHAPTER 1. OPENSHIFT SDN DEFAULT CNI NETWORK PROVIDER

1.1. ENABLING MULTICAST FOR A PROJECT

1.1.1. About multicast

With IP multicast, data is broadcast to many IP addresses simultaneously.



IMPORTANT

At this time, multicast is best used for low-bandwidth coordination or service discovery and not a high-bandwidth solution.

Multicast traffic between OpenShift Dedicated pods is disabled by default. If you are using the OpenShift SDN default Container Network Interface (CNI) network provider, you can enable multicast on a per-project basis.

When using the OpenShift SDN network plug-in in **networkpolicy** isolation mode:

- Multicast packets sent by a pod will be delivered to all other pods in the project, regardless of **NetworkPolicy** objects. Pods might be able to communicate over multicast even when they cannot communicate over unicast.
- Multicast packets sent by a pod in one project will never be delivered to pods in any other project, even if there are **NetworkPolicy** objects that allow communication between the projects.

When using the OpenShift SDN network plug-in in **multitenant** isolation mode:

- Multicast packets sent by a pod will be delivered to all other pods in the project.
- Multicast packets sent by a pod in one project will be delivered to pods in other projects only if each project is joined together and multicast is enabled in each joined project.

1.1.2. Enabling multicast between pods

You can enable multicast between pods for your project.

Prerequisites

- Install the OpenShift CLI (**oc**).
- You must log in to the cluster with a user that has the **cluster-admin** or the **dedicated-admin** role.

Procedure

- Run the following command to enable multicast for a project. Replace **<namespace>** with the namespace for the project you want to enable multicast for.

```
$ oc annotate netnamespace <namespace> \
    netnamespace.network.openshift.io/multicast-enabled=true
```

-

Verification

To verify that multicast is enabled for a project, complete the following procedure:

1. Change your current project to the project that you enabled multicast for. Replace **<project>** with the project name.

```
$ oc project <project>
```

2. Create a pod to act as a multicast receiver:

```
$ cat <<EOF| oc create -f -
apiVersion: v1
kind: Pod
metadata:
  name: mlistener
  labels:
    app: multicast-verify
spec:
  containers:
    - name: mlistener
      image: registry.access.redhat.com/ubi8
      command: ["/bin/sh", "-c"]
      args:
        ["dnf -y install socat hostname && sleep inf"]
      ports:
        - containerPort: 30102
          name: mlistener
          protocol: UDP
EOF
```

3. Create a pod to act as a multicast sender:

```
$ cat <<EOF| oc create -f -
apiVersion: v1
kind: Pod
metadata:
  name: msender
  labels:
    app: multicast-verify
spec:
  containers:
    - name: msender
      image: registry.access.redhat.com/ubi8
      command: ["/bin/sh", "-c"]
      args:
        ["dnf -y install socat && sleep inf"]
EOF
```

4. In a new terminal window or tab, start the multicast listener.

   a. Get the IP address for the Pod:

```
$ POD_IP=$(oc get pods mlistener -o jsonpath='{.status.podIP}')
```

b. Start the multicast listener by entering the following command:

```
$ oc exec mlistener -i -t -- \
    socat UDP4-RECVFROM:30102,ip-add-membership=224.1.0.1:$POD_IP,fork
EXEC:hostname
```

5. Start the multicast transmitter.

a. Get the pod network IP address range:

```
$ CIDR=$(oc get Network.config.openshift.io cluster \
    -o jsonpath='{.status.clusterNetwork[0].cidr}')
```

b. To send a multicast message, enter the following command:

```
$ oc exec msender -i -t -- \
    /bin/bash -c "echo | socat STDIO UDP4-
DATAGRAM:224.1.0.1:30102,range=$CIDR,ip-multicast-ttl=64"
```

If multicast is working, the previous command returns the following output:

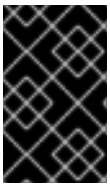
```
mlistener
```

CHAPTER 2. CONFIGURING A CLUSTER-WIDE PROXY DURING INSTALLATION

You can configure a cluster-wide proxy during cluster installation or after the cluster has been installed.

If you use a cluster-wide proxy, you are responsible for the following:

- Maintaining the availability of the proxy to the cluster.
- Understanding that if the proxy becomes unavailable, then it may impact the health and supportability of the cluster.



IMPORTANT

Cluster-wide proxy is a functionally-complete feature and suitable for production workloads. There are additional considerations that need to be added to documentation, and until then, this feature is considered a Technology Preview.

2.1. PREREQUISITES FOR CONFIGURING A CLUSTER-WIDE PROXY

To configure a cluster-wide proxy, you must meet the following requirements. These requirements are valid for both fresh installation and post installation proxy configuration.

2.1.1. General requirements

- You are the cluster owner.
- Your account has sufficient privileges.
- You have added the **ec2.<region>.amazonaws.com**, **elasticloadbalancing.<region>.amazonaws.com**, and **s3.<region>.amazonaws.com** endpoints to your virtual private cloud (VPC) endpoint. These endpoints are required to complete requests from the nodes to the AWS EC2 API. Because the proxy works on the container level, not the node level, you must route these requests to the AWS EC2 API through the AWS private network. Adding the public IP address of the EC2 API to your allowlist in your proxy server is not sufficient.
- You must have a Customer Cloud Subscription (CCS) cluster with a VPC that the proxy can access.
- You have the **ocm** CLI installed and configured.

2.1.2. Network requirements

- If your proxy re-encyrpts egress traffic, you must create exclusions to the domain and port combinations. The following table offers guidance into these exceptions.
    - Allowlist the following OpenShift URLs for re-encryption.

Address	Protocol/Port	Function
observatorium-mst.api.openshift.com	https/443	Required. Used for Managed OpenShift-specific telemetry.
sso.redhat.com	https/443	The https://cloud.redhat.com/openshift site uses authentication from sso.redhat.com to download the cluster pull secret and use Red Hat SaaS solutions to facilitate monitoring of your subscriptions, cluster inventory, and chargeback reporting.

- Allowlist the following site reliability engineering (SRE) and management URLs for re-encryption.

Address	Protocol/Port	Function
***.osdsecuritylogs.splunkcloud.com** OR **inputs1.osdsecuritylogs.splunkcloud.cominputs2.osdsecuritylogs.splunkcloud.cominputs4.osdsecuritylogs.splunkcloud.cominputs5.osdsecuritylogs.splunkcloud.cominputs6.osdsecuritylogs.splunkcloud.cominputs7.osdsecuritylogs.splunkcloud.cominputs8.osdsecuritylogs.splunkcloud.cominputs9.osdsecuritylogs.splunkcloud.cominputs10.osdsecuritylogs.splunkcloud.cominputs11.osdsecuritylogs.splunkcloud.cominputs12.osdsecuritylogs.splunkcloud.cominputs13.osdsecuritylogs.splunkcloud.cominputs14.osdsecuritylogs.splunkcloud.cominputs15.osdsecuritylogs.splunkcloud.com**	tcp/9997	Used by the splunk-forwarder-operator as a log forwarding endpoint to be used by Red Hat SRE for log-based alerting.

Address	Protocol/Port	Function
http-inputs-osdsecuritylogs.splunkcloud.com	https/443	Used by the splunk-forwarder-operator as a log forwarding endpoint to be used by Red Hat SRE for log-based alerting.

Additional Resources

For more information, see Getting started with OpenShift Dedicated for a basic cluster installation workflow.

IMPORTANT

The use of a proxy server to perform TLS re-encryption is currently not supported if the server is acting as a transparent forward proxy where it is not configured on-cluster via the **--http-proxy** or **--https-proxy** arguments.

A transparent forward proxy intercepts the cluster's traffic, but it is not actually configured on the cluster itself.

2.2. CONFIGURING A CLUSTER-WIDE PROXY DURING INSTALLATION

You can add a proxy during cluster installation. Prior to installation, however, you should verify that the proxy is accessible from the intended cluster virtual private cloud (VPC) and its private subnets.

WARNING

Only cluster system egress traffic is proxied, including calls to the AWS API. A system-wide proxy does not affect user workloads. It only affects system components.

Procedure

- To create a cluster with a proxy, run the following command:

```
$ ocm create cluster \
 <other_arguments_here> \
 --additional-trust-bundle-file <path_to_CA_bundle_file> \ 1 2 3
 --http-proxy http://<username>:<pswd>@<ip>:<port> \ 4 5
 --https-proxy http(s)://<username>:<pswd>@<ip>:<port> 6
```

1 4 The **http-proxy**, **https-proxy**, and **additional-trust-bundle-file** arguments are all optional.

2

If you use the **additional-trust-bundle-file** option without an **http(s)-proxy** argument, the passed additional trust bundle is set on the cluster, but it is not configured to be used with

3 The **additional-trust-bundle-file** argument is a file path pointing to a bundle of PEM-encoded X.509 certificates, which are all concatenated together. The **additionalTrustBundle** parameter is required unless the proxy's identity certificate is signed by an authority from the RHCOS trust bundle. If you use an MITM transparent proxy network that does not require additional proxy configuration but requires additional CAs, you must provide the MITM CA certificate.

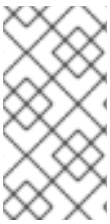
5 6 The **http-proxy** and **https-proxy** arguments must point to a valid URL.

2.3. CONFIGURING OR UPDATING YOUR CLUSTER-WIDE PROXY AFTER INSTALLATION

As the cluster owner, you may wish to add a proxy to your created cluster after installation, or you may wish to make changes to your proxy that you configured during cluster installation. The **ocm** CLI provides some options for adding a proxy to your cluster or modifying an existing proxy on your cluster.

You may need to perform these actions if:

- the cluster-wide proxy is configured after installation,
- the proxy's network address needs to be updated, and/or
- any of the proxy's certificate authorities have expired and the additional trust bundle needs to be replaced.



NOTE

The cluster applies the configuration to the cluster's control plane and worker nodes. This process results in each node in the cluster temporarily being placed into an unschedulable state and drained of its workloads while applying the configuration. Each node will be restarted as part of this process.

Procedure

- To edit a cluster, run the following command:

```
$ ocm edit cluster \
 --cluster $CLUSTER_NAME \
 --additional-trust-bundle-file $CA_BUNDLE_FILE \
 --http-proxy $HTTP_PROXY \
 --https-proxy $HTTPS_PROXY
```

  While the **additional-trust-bundle-file**, **http-proxy**, and **https-proxy** arguments are optional, if you set a **additional-trust-bundle-file** without either an **http-proxy** or **https-proxy** argument, then the additional trust bundle will still be used for verifying cluster system egress traffic.

- You can verify that the proxy and certificate authority configuration updates have been successfully applied to your cluster by:
  - All of the MachineConfigPools are updated. Run the following command to see their status:

```
$ oc get machineconfigpools
```

Sample Output

```
NAME     CONFIG                                             UPDATED   UPDATING   DEGRADED
MACHINECOUNT   READYMACHINECOUNT   UPDATEDMACHINECOUNT
DEGRADEDMACHINECOUNT   AGE
master   rendered-master-d9a03f612a432095dcde6dcf44597d90   True      False
False      3              3                 3                   0                   31h
worker   rendered-worker-f6827a4efe21e155c25c21b43c46f65e   True      False      False
6              6                 6                   0                   31h
```

- As the cluster owner, the following command displays the proxy status:

```
$ oc get proxy cluster -o yaml
```

Sample Output

```
apiVersion: config.openshift.io/v1
kind: Proxy
spec:
  httpProxy: http://proxy.host.domain:<port>
  httpsProxy: https://proxy.host.domain:<port>
  <...more...>
status:
  httpProxy: http://proxy.host.domain:<port>
  httpsProxy: https://proxy.host.domain:<port>
  <...more...>
```

NOTE

You should not attempt to change the proxy or additional trust bundle configuration on the cluster itself. These changes should always be done via the **ocm** command-line tools. Any changes that are made directly to the cluster will be reverted automatically.

2.3.1. Responsibilites for additional trust bundles

If you supplied an additional trust bundle file, you are responsible for the following:

- Ensuring that the contents of the additional trust bundle are valid,
- Ensuring that the certificates, including intermediary certificates, contained in the additional trust bundle have not expired, and
- Tracking the expiry and performing any necessary renewals for certificates contained in the additional trust bundle, and subsequently updating the cluster's configuration with the updated additional trust bundle.

CHAPTER 3. CIDR RANGE DEFINITIONS

You must specify non-overlapping ranges for the following CIDR ranges.

NOTE

Machine CIDR ranges cannot be changed after creating your cluster.

3.1. MACHINE CIDR

In the Machine CIDR field, you must specify the IP address range for machines or cluster nodes. This range must encompass all CIDR address ranges for your virtual private cloud (VPC) subnets. Subnets must be contiguous. A minimum IP address range of 128 addresses, using the subnet prefix **/25**, is supported for single availability zone deployments. A minimum address range of 256 addresses, using the subnet prefix **/24**, is supported for deployments that use multiple availability zones. The default is **10.0.0.0/16**. This range must not conflict with any connected networks.

3.2. SERVICE CIDR

In the Service CIDR field, you must specify the IP address range for services. The range must be large enough to accommodate your workload. The address block must not overlap with any external service accessed from within the cluster. The default is **172.30.0.0/16**. This address block needs to be the same between clusters.

3.3. POD CIDR

In the pod CIDR field, you must specify the IP address range for pods. The range must be large enough to accommodate your workload. The address block must not overlap with any external service accessed from within the cluster. The default is **10.128.0.0/14**. This address block needs to be the same between clusters.

3.4. HOST PREFIX

In the Host Prefix field, you must Specify the subnet prefix length assigned to pods scheduled to individual machines. The host prefix determines the pod IP address pool for each machine. For example, if the host prefix is set to **/23**, each machine is assigned a **/23** subnet from the pod CIDR address range. The default is **/23**, allowing 512 cluster nodes, and 512 pods per node (both of which are beyond our maximum supported).