



# OpenShift Container Platform 4.2

## Installing

Installing and configuring OpenShift Container Platform 4.2 clusters



# OpenShift Container Platform 4.2 Installing

---

Installing and configuring OpenShift Container Platform 4.2 clusters

## Legal Notice

Copyright © 2020 Red Hat, Inc.

The text of and illustrations in this document are licensed by Red Hat under a Creative Commons Attribution–Share Alike 3.0 Unported license ("CC-BY-SA"). An explanation of CC-BY-SA is available at

<http://creativecommons.org/licenses/by-sa/3.0/>

. In accordance with CC-BY-SA, if you distribute this document or an adaptation of it, you must provide the URL for the original version.

Red Hat, as the licensor of this document, waives the right to enforce, and agrees not to assert, Section 4d of CC-BY-SA to the fullest extent permitted by applicable law.

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, the Red Hat logo, JBoss, OpenShift, Fedora, the Infinity logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

Linux<sup>®</sup> is the registered trademark of Linus Torvalds in the United States and other countries.

Java<sup>®</sup> is a registered trademark of Oracle and/or its affiliates.

XFS<sup>®</sup> is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL<sup>®</sup> is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js<sup>®</sup> is an official trademark of Joyent. Red Hat is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack<sup>®</sup> Word Mark and OpenStack logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

## Abstract

This document provides information about installing OpenShift Container Platform 4.2 and details about some configuration processes.

---

## Table of Contents

<b>CHAPTER 1. GATHERING INSTALLATION LOGS</b> .....	<b>3</b>
1.1. GATHERING LOGS FROM A FAILED INSTALLATION	3
1.2. MANUALLY GATHERING LOGS WITH SSH ACCESS TO YOUR HOST(S)	4
1.3. MANUALLY GATHERING LOGS WITHOUT SSH ACCESS TO YOUR HOST(S)	5
<b>CHAPTER 2. INSTALLATION CONFIGURATION</b> .....	<b>6</b>
2.1. INSTALLATION METHODS FOR DIFFERENT PLATFORMS	6
2.2. CREATING A MIRROR REGISTRY FOR INSTALLATION IN A RESTRICTED NETWORK	6
2.2.1. About the mirror registry	6
2.2.2. Preparing the mirror host	7
2.2.2.1. Installing the CLI	7
2.2.2.1.1. Installing the CLI on Linux	7
2.2.2.1.2. Installing the CLI on Windows	8
2.2.2.1.3. Installing the CLI on macOS	8
2.2.3. Creating a mirror registry	8
2.2.4. Adding the registry to your pull secret	11
2.2.5. Mirroring the OpenShift Container Platform image repository	13
2.2.6. Using Samples Operator imagestreams with alternate or mirrored registries	14
2.3. AVAILABLE CLUSTER CUSTOMIZATIONS	16
2.3.1. Cluster configuration resources	16
2.3.2. Operator configuration resources	17
2.3.3. Additional configuration resources	18
2.3.4. Informational Resources	18
2.4. CONFIGURING YOUR FIREWALL	19
2.4.1. Configuring your firewall for OpenShift Container Platform	19



# CHAPTER 1. GATHERING INSTALLATION LOGS

To assist in troubleshooting a failed OpenShift Container Platform installation, you can gather logs from the bootstrap and control plane, or master, machines.

## Prerequisites

- You attempted to install an OpenShift Container Platform cluster, and installation failed.
- You provided an SSH key to the installation program, and that key is in your running **ssh-agent** process.

## 1.1. GATHERING LOGS FROM A FAILED INSTALLATION

If you gave an SSH key to your installation program, you can gather data about your failed installation.



### NOTE

You use a different command to gather logs about an unsuccessful installation than to gather logs from a running cluster. If you must gather logs from a running cluster, use the **oc adm must-gather** command.

## Prerequisites

- Your OpenShift Container Platform installation failed before the bootstrap process finished. The bootstrap node must be running and accessible through SSH.
- The **ssh-agent** process is active on your computer, and you provided both the **ssh-agent** process and the installation program the same SSH key.
- If you tried to install a cluster on infrastructure that you provisioned, you must have the fully-qualified domain names of the control plane, or master, machines.

## Procedure

1. Generate the commands that are required to obtain the installation logs from the bootstrap and control plane machines:

- If you used installer-provisioned infrastructure, run the following command:

```
$ ./openshift-install gather bootstrap --dir=<directory> 1
```

- 1 **installation\_directory** is the directory you stored the OpenShift Container Platform definition files that the installation program creates.

For installer-provisioned infrastructure, the installation program stores information about the cluster, so you do not specify the host names or IP addresses

- If you used infrastructure that you provisioned yourself, run the following command:

```
$ ./openshift-install gather bootstrap --dir=<directory> \ 1
--bootstrap <bootstrap_address> \ 2
--master <master_1_address> \ 3
```

```
--master <master_2_address> \ 4
--master <master_3_address>" 5
```

- 1 **installation\_directory** is the directory you stored the OpenShift Container Platform definition files that the installation program creates.
- 2 **<bootstrap\_address>** is the fully-qualified domain name or IP address of the cluster's bootstrap machine.
- 3 4 5 For each control plane, or master, machine in your cluster, replace **<master\_\*\_address>** with its fully-qualified domain name or IP address.



#### NOTE

A default cluster contains three control plane machines. List all of your control plane machines as shown, no matter how many your cluster uses.

The command output resembles the following example:

```
INFO Pulling debug logs from the bootstrap machine
INFO Bootstrap gather logs captured here "<directory>/log-bundle-<timestamp>.tar.gz"
```

If you open a Red Hat support case about your installation failure, include the compressed logs in the case.

## 1.2. MANUALLY GATHERING LOGS WITH SSH ACCESS TO YOUR HOST(S)

Manually gather logs in situations where **must-gather** or automated collection methods do not work.

### Prerequisites

- You must have SSH access to your host(s).

### Procedure

1. Collect the **bootkube.service** service logs from the bootstrap host using the **journalctl** command by running:

```
$ journalctl -b -f -u bootkube.service
```

2. Collect the bootstrap host's container logs using the Podman logs. This is shown as a loop to get all of the container logs from the host:

```
$ for pod in $(sudo podman ps -a -q); do sudo podman logs $pod; done
```

3. Alternatively, collect the host's container logs using the **tail** command by running:

```
# tail -f /var/lib/containers/storage/overlay-containers/*/userdata/ctr.log
```



4. Collect the **kubelet.service** and **crio.service** service logs from the master and worker hosts using the **journalctl** command by running:

```
$ journalctl -b -f -u kubelet.service -u crio.service
```

5. Collect the master and worker host container logs using the **tail** command by running:

```
$ sudo tail -f /var/log/containers/*
```

### 1.3. MANUALLY GATHERING LOGS WITHOUT SSH ACCESS TO YOUR HOST(S)

Manually gather logs in situations where **must-gather** or automated collection methods do not work.

If you do not have SSH access to your node, you can access the systems journal to investigate what is happening on your host.

#### Prerequisites

- Your OpenShift Container Platform installation must be complete.
- Your API service is still functional.
- You have system administrator privileges.

#### Procedure

1. Access **journal** unit logs under **/var/log** by running:

```
$ oc adm node-logs --role=master -u kubelet
```

2. Access host file paths under **/var/log** by running:

```
$ oc adm node-logs --role=master --path=openshift-apiserver
```

## CHAPTER 2. INSTALLATION CONFIGURATION

### 2.1. INSTALLATION METHODS FOR DIFFERENT PLATFORMS

You can perform different types of installations on different platforms.

Table 2.1. Installer-provisioned infrastructure options

	AWS	Azure	GCP	OpenStack	Bare metal	vSphere	IBM Z
Default	X	X	X				
Custom	X	X	X	X			
Network Operator	X	X	X				

Table 2.2. User-provisioned infrastructure options

	AWS	Azure	GCP	OpenStack	Bare metal	vSphere	IBM Z
Custom	X		X		X	X	X
Network Operator					X	X	
Restricted network	X				X	X	

### 2.2. CREATING A MIRROR REGISTRY FOR INSTALLATION IN A RESTRICTED NETWORK

Before you install a cluster on infrastructure that you provision in a restricted network, you must mirror the required container images into that environment. Installations on a restricted network are supported on only infrastructure that you provision, not infrastructure that the installer provisions.



#### IMPORTANT

You must have access to the internet to obtain the necessary container images. In this procedure, you place the mirror registry on a mirror host that has access to both your network and the internet. If you do not have access to a mirror host, use the method that best fits your restrictions to bring the contents of the mirror registry into your restricted network.

#### 2.2.1. About the mirror registry

You can mirror the contents of the OpenShift Container Platform registry and the images that are required to generate the installation program.

The mirror registry is a key component that is required to complete an installation in a restricted network. You can create this mirror on a bastion host, which can access both the internet and your closed network, or by using other methods that meet your restrictions.

Because of the way that OpenShift Container Platform verifies integrity for the release payload, the image references in your local registry are identical to the ones that are hosted by Red Hat on [Quay.io](https://quay.io). During the bootstrapping process of installation, the images must have the same digests no matter which repository they are pulled from. To ensure that the release payload is identical, you mirror the images to your local repository.

## 2.2.2. Preparing the mirror host

Before you create the mirror registry, you must prepare the mirror host.

### 2.2.2.1. Installing the CLI

You can install the OpenShift CLI (**oc**) in order to interact with OpenShift Container Platform from a command-line interface. You can install **oc** on Linux, Windows, or macOS.



#### IMPORTANT

If you installed an earlier version of **oc**, you cannot use it to complete all of the commands in OpenShift Container Platform 4.2. Download and install the new version of **oc**.

#### 2.2.2.1.1. Installing the CLI on Linux

You can install the OpenShift CLI (**oc**) binary on Linux by using the following procedure.

##### Procedure

1. Navigate to the [Infrastructure Provider](#) page on the Red Hat OpenShift Cluster Manager site.
2. Select your infrastructure provider, and, if applicable, your installation type.
3. In the **Command-line interface** section, select **Linux** from the drop-down menu and click **Download command-line tools**.
4. Unpack the archive:

```
$ tar xvzf <file>
```

5. Place the **oc** binary in a directory that is on your **PATH**.  
To check your **PATH**, execute the following command:

```
$ echo $PATH
```

After you install the CLI, it is available using the **oc** command:

```
$ oc <command>
```

### 2.2.2.1.2. Installing the CLI on Windows

You can install the OpenShift CLI (**oc**) binary on Windows by using the following procedure.

#### Procedure

1. Navigate to the [Infrastructure Provider](#) page on the Red Hat OpenShift Cluster Manager site.
2. Select your infrastructure provider, and, if applicable, your installation type.
3. In the **Command-line interface** section, select **Windows** from the drop-down menu and click **Download command-line tools**.
4. Unzip the archive with a ZIP program.
5. Move the **oc** binary to a directory that is on your **PATH**.  
To check your **PATH**, open the command prompt and execute the following command:

```
C:\> path
```

After you install the CLI, it is available using the **oc** command:

```
C:\> oc <command>
```

### 2.2.2.1.3. Installing the CLI on macOS

You can install the OpenShift CLI (**oc**) binary on macOS by using the following procedure.

#### Procedure

1. Navigate to the [Infrastructure Provider](#) page on the Red Hat OpenShift Cluster Manager site.
2. Select your infrastructure provider, and, if applicable, your installation type.
3. In the **Command-line interface** section, select **MacOS** from the drop-down menu and click **Download command-line tools**.
4. Unpack and unzip the archive.
5. Move the **oc** binary to a directory on your PATH.  
To check your **PATH**, open a terminal and execute the following command:

```
$ echo $PATH
```

After you install the CLI, it is available using the **oc** command:

```
$ oc <command>
```

## 2.2.3. Creating a mirror registry

Create a registry to host the mirrored content that you require for installing OpenShift Container Platform.

For installation in a restricted network, you can place the mirror registry on a host that can be accessed from both the your network and the internet. If you do not have access to such a host, use the method that best fits your restrictions to bring the contents of the mirror registry into your restricted network.



## NOTE

The following procedure creates a simple registry that stores data in the `/opt/registry` folder and runs in a **podman** container. You can use a different registry solution, such as [Red Hat Quay](#). Review the following procedure to ensure that your registry functions correctly.

## Prerequisites

- You have a Red Hat Enterprise Linux (RHEL) server on your network to use as the registry host.
- The registry host can access the internet.

## Procedure

On the mirror host, take the following actions:

1. Install the required packages:

```
# yum -y install podman httpd-tools
```

The **podman** package provides the container package that you run the registry in. The **httpd-tools** package provides the **htpasswd** utility, which you use to create users.

2. Create folders for the registry:

```
# mkdir -p /opt/registry/{auth,certs,data}
```

These folders are mounted inside the registry container.

3. Provide a certificate for the registry. If you do not have an existing, trusted certificate authority, you can generate a self-signed certificate:

```
$ cd /opt/registry/certs
# openssl req -newkey rsa:4096 -nodes -sha256 -keyout domain.key -x509 -days 365 -out domain.crt
```

At the prompts, provide the required values for the certificate:

Country Name (2 letter code)	Specify the two-letter ISO country code for your location. See the <a href="#">ISO 3166 country codes</a> standard.
State or Province Name (full name)	Enter the full name of your state or province.

Locality Name (eg, city)	Enter the name of your city.
Organization Name (eg, company)	Enter your company name.
Organizational Unit Name (eg, section)	Enter your department name.
Common Name (eg, your name or your server's hostname)	Enter the host name for the registry host. Ensure that your hostname is in DNS and that it resolves to the expected IP address.
Email Address	Enter your email address. For more information, see the <a href="#">req</a> description in the OpenSSL documentation.

4. Generate a user name and a password for your registry that uses the **bcrpt** format:

```
# htpasswd -bBc /opt/registry/auth/htpasswd <user_name> <password> ❶
```

- ❶ Replace **<user\_name>** and **<password>** with a user name and a password.

5. Create the **mirror-registry** container to host your registry:

```
# podman run --name mirror-registry -p <local_registry_host_port>:5000 \ ❶
-v /opt/registry/data:/var/lib/registry:z \
-v /opt/registry/auth:/auth:z \
-e "REGISTRY_AUTH=htpasswd" \
-e "REGISTRY_AUTH_HTPASSWD_REALM=Registry Realm" \
-e REGISTRY_AUTH_HTPASSWD_PATH=/auth/htpasswd \
-v /opt/registry/certs:/certs:z \
-e REGISTRY_HTTP_TLS_CERTIFICATE=/certs/domain.crt \
-e REGISTRY_HTTP_TLS_KEY=/certs/domain.key \
-e REGISTRY_COMPATIBILITY_SCHEMA1_ENABLED=true \
-d docker.io/library/registry:2
```

- ❶ For **<local\_registry\_host\_port>**, specify the port that your mirror registry uses to serve content.

6. Open the required ports for your registry:

```
# firewall-cmd --add-port=<local_registry_host_port>/tcp --zone=internal --permanent 1
# firewall-cmd --add-port=<local_registry_host_port>/tcp --zone=public --permanent 2
# firewall-cmd --reload
```

- 1** **2** For **<local\_registry\_host\_port>**, specify the port that your mirror registry uses to serve content.

7. Add the self-signed certificate to your list of trusted certificates:

```
# cp /opt/registry/certs/domain.crt /etc/pki/ca-trust/source/anchors/
# update-ca-trust
```

You must trust your certificate to log in to your registry during the mirror process.

8. Confirm that the registry is available:

```
$ curl -u <user_name>:<password> -k https://<local_registry_host_name>:
<local_registry_host_port>/v2/_catalog 1

{"repositories":[]}
```

- 1** For **<user\_name>** and **<password>**, specify the user name and password for your registry. For **<local\_registry\_host\_name>**, specify the registry domain name that you specified in your certificate, such as **registry.example.com**. For **<local\_registry\_host\_port>**, specify the port that your mirror registry uses to serve content.

If the command output displays an empty repository, your registry is available.

## 2.2.4. Adding the registry to your pull secret

Modify your the pull secret for your OpenShift Container Platform cluster to describe your local registry before you install an OpenShift Container Platform cluster in a restricted network.

### Prerequisites

- You configured a mirror registry to use in your restricted network.

### Procedure

Complete the following steps on the mirror host:

- Download your **registry.redhat.io** pull secret from the [Pull Secret](#) page on the Red Hat OpenShift Cluster Manager site.
- Generate the base64-encoded user name and password or token for your mirror registry:

```
$ echo -n '<user_name>:<password>' | base64 -w0 1
BGVtbYk3ZHAtdXs=
```

- 1** For **<user\_name>** and **<password>**, specify the user name and password that you configured for your registry.

3. Make a copy of your pull secret in JSON format:

```
$ cat ./pull-secret.text | jq . > <path>/<pull-secret-file> 1
```

- 1** Specify the path to the folder to store the pull secret in and a name for the JSON file that you create.

The contents of the file resemble the following example:

```
{
  "auths": {
    "cloud.openshift.com": {
      "auth": "b3BlbnNo...",
      "email": "you@example.com"
    },
    "quay.io": {
      "auth": "b3BlbnNo...",
      "email": "you@example.com"
    },
    "registry.connect.redhat.com": {
      "auth": "NTE3Njg5Nj...",
      "email": "you@example.com"
    },
    "registry.redhat.io": {
      "auth": "NTE3Njg5Nj...",
      "email": "you@example.com"
    }
  }
}
```

4. Edit the new file and add a section that describes your registry to it:

```
"auths": {
  ...
  "<mirror_registry>": { 1
    "auth": "<credentials>", 2
    "email": "you@example.com"
  },
  ...
}
```

- 1** For **<mirror\_registry>**, specify the registry domain name, and optionally the port, that your mirror registry uses to serve content. For example, **registry.example.com** or **registry.example.com:5000**
- 2** For **<credentials>**, specify the base64-encoded user name and password for the mirror registry.

The file resembles the following example:

```
{
  "auths": {
    "cloud.openshift.com": {
      "auth": "b3BlbnNo...",
```



```

    "email": "you@example.com"
  },
  "quay.io": {
    "auth": "b3BlbnNo...",
    "email": "you@example.com"
  },
  "registry.connect.redhat.com": {
    "auth": "NTE3Njg5Nj...",
    "email": "you@example.com"
  },
  "<mirror_registry>": {
    "auth": "<credentials>",
    "email": "you@example.com"
  },
  "registry.redhat.io": {
    "auth": "NTE3Njg5Nj...",
    "email": "you@example.com"
  }
}
}
}

```

## 2.2.5. Mirroring the OpenShift Container Platform image repository

Mirror the OpenShift Container Platform image repository to use during cluster installation or upgrade.

### Prerequisites

- You configured a mirror registry to use in your restricted network and can access the certificate and credentials that you configured.
- You downloaded the pull secret from the [Pull Secret](#) page on the Red Hat OpenShift Cluster Manager site and modified it to include authentication to your mirror repository.

### Procedure

Complete the following steps on a host that can access both quay.io and your mirror registry:

1. Review the [OpenShift Container Platform downloads page](#) to determine the version of OpenShift Container Platform that you want to install.
2. Set the required environment variables:

```

$ export OCP_RELEASE=<release_version> 1
$ export LOCAL_REGISTRY=<local_registry_host_name>:<local_registry_host_port>' 2
$ export LOCAL_REPOSITORY=<repository_name>' 3
$ export PRODUCT_REPO='openshift-release-dev' 4
$ export LOCAL_SECRET_JSON=<path_to_pull_secret>' 5
$ export RELEASE_NAME="ocp-release" 6

```

- 1 For **<release\_version>**, specify the version number of OpenShift Container Platform to install, such as **4.2.0**.
- 2 For **<local\_registry\_host\_name>**, specify the registry domain name for your mirror repository, and for **<local\_registry\_host\_port>**, specify the port that it serves content on.

- 3 For **<repository\_name>**, specify the name of the repository to create in your registry, such as **ocp4/openshift4**.
- 4 The repository to mirror. For a production release, you must specify **openshift-release-dev**.
- 5 For **<path\_to\_pull\_secret>**, specify the absolute path to and file name of the pull secret for your mirror registry that you created.
- 6 The release mirror. For a production release, you must specify **ocp-release**.

### 3. Mirror the repository:

```
$ oc adm -a ${LOCAL_SECRET_JSON} release mirror \
  --from=quay.io/${PRODUCT_REPO}/${RELEASE_NAME}:${OCP_RELEASE} \
  --to=${LOCAL_REGISTRY}/${LOCAL_REPOSITORY} \
  --to-release-image=${LOCAL_REGISTRY}/${LOCAL_REPOSITORY}:${OCP_RELEASE}
```

This command pulls the release information as a digest, and its output includes the **imageContentSources** data that you require when you install your cluster.

4. Record the entire **imageContentSources** section from the output of the previous command. The information about your mirrors is unique to your mirrored repository, and you must add the **imageContentSources** section to the **install-config.yaml** file during installation.
5. To create the installation program that is based on the content that you mirrored, extract it and pin it to the release:

```
$ oc adm -a ${LOCAL_SECRET_JSON} release extract --command=openshift-install
"${LOCAL_REGISTRY}/${LOCAL_REPOSITORY}:${OCP_RELEASE}"
```



#### IMPORTANT

To ensure that you use the correct images for the version of OpenShift Container Platform that you selected, you must extract the installation program from the mirrored content.

You must perform this step on a machine with an active internet connection.

### 2.2.6. Using Samples Operator imagestreams with alternate or mirrored registries

Most imagestreams in the OpenShift namespace managed by the Samples Operator point to images located in the Red Hat registry at [registry.redhat.io](https://registry.redhat.io). Mirroring will not apply to these imagestreams.



## IMPORTANT

The **jenkins**, **jenkins-agent-maven**, and **jenkins-agent-nodejs** imagestreams come from the install payload and are managed by the Samples Operator, so no further mirroring procedures are needed for those imagestreams.

Setting the **samplesRegistry** field in the Sample Operator configuration file to [registry.redhat.io](https://registry.redhat.io) is redundant because it is already directed to [registry.redhat.io](https://registry.redhat.io) for everything but Jenkins images and imagestreams. It also breaks the installation payload for Jenkins imagestreams.

The Samples Operator prevents the use of the following registries for the Jenkins imagestreams:

- [docker.io](https://docker.io)
- [registry.redhat.io](https://registry.redhat.io)
- [registry.access.redhat.com](https://registry.access.redhat.com)
- [quay.io](https://quay.io).



## NOTE

The **cli**, **installer**, **must-gather**, and **tests** imagestreams, while part of the install payload, are not managed by the Samples Operator. These are not addressed in this procedure.

## Prerequisites

- Access to the cluster as a user with the **cluster-admin** role.
- Create a pull secret for your mirror registry.

## Procedure

1. Access the images of a specific imagestream to mirror, for example:

```
$ oc get is <imagestream> -n openshift -o json | jq .spec.tags[].from.name | grep registry.redhat.io
```

2. Mirror images from [registry.redhat.io](https://registry.redhat.io) associated with any imagestreams you need in the restricted network environment into one of the defined mirrors, for example:

```
$ oc image mirror registry.redhat.io/rhsc/ruby-25-rhel7:latest ${MIRROR_ADDR}/rhsc/ruby-25-rhel7:latest
```

3. Add the required trusted CAs for the mirror in the cluster's image configuration object:

```
$ oc create configmap registry-config --from-file=${MIRROR_ADDR_HOSTNAME}..5000=$path/ca.crt -n openshift-config
$ oc patch image.config.openshift.io/cluster --patch '{"spec":{"additionalTrustedCA":{"name":"registry-config"}}}' --type=merge
```

4. Update the **samplesRegistry** field in the Samples Operator configuration object to contain the **hostname** portion of the mirror location defined in the mirror configuration:

```
$ oc get configs.samples.operator.openshift.io -n openshift-cluster-samples-operator
```

**NOTE**

This is required because the imagestream import process does not use the mirror or search mechanism at this time.

5. Add any imagestreams that are not mirrored into the **skippedImagestreams** field of the Samples Operator configuration object. Or if you do not want to support any of the sample imagestreams, set the Samples Operator to **Removed** in the Samples Operator configuration object.

**NOTE**

Any unmirrored imagestreams that are not skipped, or if the Samples Operator is not changed to **Removed**, will result in the Samples Operator reporting a **Degraded** status two hours after the imagestream imports start failing.

Many of the templates in the OpenShift namespace reference the imagestreams. So using **Removed** to purge both the imagestreams and templates will eliminate the possibility of attempts to use them if they are not functional because of any missing imagestreams.

**Next steps**

- Install a cluster on infrastructure that you provision in your restricted network, such as on [VMware vSphere](#), [bare metal](#), or [Amazon Web Services](#).

## 2.3. AVAILABLE CLUSTER CUSTOMIZATIONS

You complete most of the cluster configuration and customization after you deploy your OpenShift Container Platform cluster. A number of *configuration resources* are available.

**NOTE**

If you install your cluster on IBM Z, not all features and functions are available. See the [release notes](#) for details.

You modify the configuration resources to configure the major features of the cluster, such as the image registry, networking configuration, image build behavior, and the identity provider.

For current documentation of the settings that you control by using these resources, use the **oc explain** command, for example **oc explain builds --api-version=config.openshift.io/v1**

### 2.3.1. Cluster configuration resources

All cluster configuration resources are globally scoped (not namespaced) and named **cluster**.

Resource name	Description
---------------	-------------

Resource name	Description
apiserver.config.openshift.io	Provides api-server configuration such as <a href="#">certificates and certificate authorities</a> .
authentication.config.openshift.io	Controls the <a href="#">identity provider</a> and authentication configuration for the cluster.
build.config.openshift.io	Controls default and enforced <a href="#">configuration</a> for all builds on the cluster.
console.config.openshift.io	Configures the behavior of the web console interface, including the <a href="#">logout behavior</a> .
featuregate.config.openshift.io	Enables <a href="#">FeatureGates</a> so that you can use Tech Preview features.
image.config.openshift.io	Configures how specific <a href="#">image registries</a> should be treated (allowed, disallowed, insecure, CA details).
ingress.config.openshift.io	Configuration details related to <a href="#">routing</a> such as the default domain for routes.
oauth.config.openshift.io	Configures identity providers and other behavior related to <a href="#">internal OAuth server</a> flows.
project.config.openshift.io	Configures <a href="#">how projects are created</a> including the project template.
proxy.config.openshift.io	Defines proxies to be used by components needing external network access. Note: not all components currently consume this value.
scheduler.config.openshift.io	Configures <a href="#">scheduler</a> behavior such as policies and default nodeselectors.

### 2.3.2. Operator configuration resources

These configuration resources are cluster-scoped instances, named **cluster**, which control the behavior of a specific component as owned by a particular operator.

Resource name	Description
console.operator.openshift.io	Controls console appearance such as branding customizations
config.imageregistry.operator.openshift.io	Configures <a href="#">internal image registry settings</a> such as public routing, log levels, proxy settings, resource constraints, replica counts, and storage type.

Resource name	Description
config.samples.operator.openshift.io	Configures the <a href="#">Samples Operator</a> to control which example imagestreams and templates are installed on the cluster.

### 2.3.3. Additional configuration resources

These configuration resources represent a single instance of a particular component. In some cases, you can request multiple instances by creating multiple instances of the resource. In other cases, the Operator can use only a specific resource instance name in a specific namespace. Reference the component-specific documentation for details on how and when you can create additional resource instances.

Resource name	Instance name	Namespace	Description
alertmanager.monitoring.coreos.com	main	openshift-monitoring	Controls the <a href="#">alertmanager</a> deployment parameters.
ingresscontroller.operator.openshift.io	default	openshift-ingress-operator	Configures <a href="#">Ingress Operator</a> behavior such as domain, number of replicas, certificates, and controller placement.

### 2.3.4. Informational Resources

You use these resources to retrieve information about the cluster. Do not edit these resources directly.

Resource name	Instance name	Description
clusterversion.config.openshift.io	version	In OpenShift Container Platform 4.2, you must not customize the ClusterVersion resource for production clusters. Instead, follow the process to <a href="#">update a cluster</a> .
dns.config.openshift.io	cluster	You cannot modify the DNS settings for your cluster. You can <a href="#">view the DNS Operator status</a> .
infrastructure.config.openshift.io	cluster	Configuration details allowing the cluster to interact with its cloud provider.
network.config.openshift.io	cluster	You cannot modify your cluster networking after installation. To customize your network, follow the process to <a href="#">customize networking during installation</a> .

## 2.4. CONFIGURING YOUR FIREWALL

If you use a firewall, you must configure it so that OpenShift Container Platform can access the sites that it requires to function. You must always grant access to some sites, and you grant access to more if you use Red Hat Insights, the Telemetry service, a cloud to host your cluster, and certain build strategies.

### 2.4.1. Configuring your firewall for OpenShift Container Platform

Before you install OpenShift Container Platform, you must configure your firewall to grant access to the sites that OpenShift Container Platform requires.

#### Procedure

1. Whitelist the following registry URLs:

URL	Function
<b>registry.redhat.io</b>	Provides core container images
<b>quay.io</b>	Provides core container images
<b>sso.redhat.com</b>	The <a href="https://cloud.redhat.com/openshift">https://cloud.redhat.com/openshift</a> site uses authentication from <b>sso.redhat.com</b>

2. Whitelist any site that provides resources for a language or framework that your builds require.
3. If you do not disable Telemetry, you must grant access to the following URLs to access Red Hat Insights:

URL	Function
<b>cert-api.access.redhat.com</b>	Required for Telemetry
<b>api.access.redhat.com</b>	Required for Telemetry
<b>infogw.api.openshift.com</b>	Required for Telemetry
<a href="https://cloud.redhat.com/api/ingress">https://cloud.redhat.com/api/ingress</a>	Required for Telemetry and for <b>insights-operator</b>

4. If you use Amazon Web Services (AWS), Microsoft Azure, or Google Cloud Platform (GCP) to host your cluster, you must grant access to the URLs that provide the cloud provider API and DNS for that cloud:

Cloud	URL	Function
-------	-----	----------

Cloud	URL	Function
AWS	<b>*.amazonaws.com</b>	Required to access AWS services and resources. Review the <a href="#">AWS Service Endpoints</a> in the AWS documentation to determine the exact endpoints to allow for the regions that you use.
GCP	<b>*.googleapis.com</b>	Required to access GCP services and resources. Review <a href="#">Cloud Endpoints</a> in the GCP documentation to determine the endpoints to allow for your APIs.
	<b>accounts.google.com</b>	Required to access your GCP account.
Azure	<b>management.azure.com</b>	Required to access Azure services and resources. Review the <a href="#">Azure REST API Reference</a> in the Azure documentation to determine the endpoints to allow for your APIs.

## 5. Whitelist the following URLs:

URL	Function
<b>mirror.openshift.com</b>	Required to access mirrored installation content and images. This site is also a source of release image signatures, although the Cluster Version Operator needs only a single functioning source.
<b>storage.googleapis.com/openshift-release</b>	A source of release image signatures, although the Cluster Version Operator needs only a single functioning source.
<b>*.apps.&lt;cluster_name&gt;.&lt;base_domain&gt;</b>	Required to access the default cluster routes unless you set an ingress wildcard during installation.
<b>quay-registry.s3.amazonaws.com</b>	Required to access Quay image content in AWS.
<b>api.openshift.com</b>	Required to check if updates are available for the cluster.
<b>art-rhcos-ci.s3.amazonaws.com</b>	Required to download Red Hat Enterprise Linux CoreOS (RHCOS) images.
<b>api.openshift.com</b>	Required for your cluster token.
<b>cloud.redhat.com/openshift</b>	Required for your cluster token.



