OpenShift Container Platform 4.3

Installing on OpenStack

Installing OpenShift Container Platform 4.3 OpenStack clusters

Last Updated: 2020-04-08

OpenShift Container Platform 4.3 Installing on OpenStack

Installing OpenShift Container Platform 4.3 OpenStack clusters

Legal Notice

Abstract

This document provides instructions for installing and uninstalling OpenShift Container Platform 4.3 clusters on OpenStack Container Platform.

Table of Contents

CHAPTER 1. INSTALLING ON OPENSTACK

1.1. INSTALLING A CLUSTER ON OPENSTACK WITH CUSTOMIZATIONS

In OpenShift Container Platform version 4.3, you can install a customized cluster on Red Hat OpenStack Platform (RHOSP). To customize the installation, modify parameters in the **install-config.yaml** before you install the cluster.

Prerequisites

- Review details about the OpenShift Container Platform installation and update processes.
- Have access to a RHOSP administrator's account
- Have metadata service enabled in RHOSP

1.1.1. Resource guidelines for installing OpenShift Container Platform on OpenStack

Your quota must meet the following requirements to run the OpenShift Container Platform installation program in Red Hat OpenStack Platform (RHOSP).

Table 1.1. Recommended resources for a default OpenShift Container Platform cluster on RHOSP

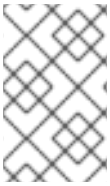
Resource	Value
Floating IP addresses	2
Ports	15
Routers	1
Subnets	1
RAM	112 GB
vCPUs	28
Volume storage	175 GB
Instances	7
Security groups	3
Security group rules	60
Swift containers	2
Swift objects	1

Resource	Value
Swift available space	10 MB or more

NOTE

Swift space requirements vary depending on the size of the bootstrap Ignition file and image registry.

A cluster might function with fewer than recommended resources, but its performance is not guaranteed.



NOTE

By default, your security group and security group rule quotas might be low. If you encounter problems, run **openstack quota set --secgroups 3 --secgroup-rules 60 <project>** to increase them.

An OpenShift Container Platform deployment comprises control plane machines, compute machines, and a bootstrap machine.

1.1.1.1. Control plane and compute machines

By default, the OpenShift Container Platform installation program stands up three control plane and compute machines.

Each machine requires:

- An instance from the RHOSP quota
- A port from the RHOSP quota
- A flavor with at least 16 GB memory, 4 vCPUs, and 25 GB storage space

TIP

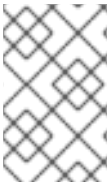
Compute machines host the applications that you run on OpenShift Container Platform; aim to run as many as you can.

1.1.1.2. Bootstrap machine

During installation, a bootstrap machine is temporarily provisioned to stand up the control plane. After the production control plane is ready, the bootstrap machine is deprovisioned.

The bootstrap machine requires:

- An instance from the RHOSP quota
- A port from the RHOSP quota
- A flavor with at least 16 GB memory, 4 vCPUs, and 25 GB storage space



NOTE

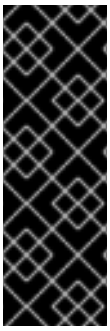
The installation program cannot pass certificate authority bundles to Ignition on control plane machines. Therefore, the bootstrap machine cannot retrieve Ignition configurations from Swift if your endpoint uses self-signed certificates.

1.1.2. Internet and Telemetry access for OpenShift Container Platform

In OpenShift Container Platform 4.3, you require access to the internet to install and entitle your cluster. The Telemetry service, which runs by default to provide metrics about cluster health and the success of updates, also requires internet access. If your cluster is connected to the internet, Telemetry runs automatically, and your cluster is registered to the Red Hat OpenShift Cluster Manager . From there, you can allocate entitlements to your cluster.

You must have internet access to:

- Access the Red Hat OpenShift Cluster Manager page to download the installation program and perform subscription management and entitlement. If the cluster has internet access and you do not disable Telemetry, that service automatically entitles your cluster. If the Telemetry service cannot entitle your cluster, you must manually entitle it on the Cluster registration page.
- Access Quay.io to obtain the packages that are required to install your cluster.
- Obtain the packages that are required to perform cluster updates.



IMPORTANT

If your cluster cannot have direct internet access, you can perform a restricted network installation on some types of infrastructure that you provision. During that process, you download the content that is required and use it to populate a mirror registry with the packages that you need to install a cluster and generate the installation program. With some installation types, the environment that you install your cluster in will not require internet access. Before you update the cluster, you update the content of the mirror registry.

1.1.3. Enabling Swift on OpenStack

OpenShift Container Platform on Red Hat OpenStack Platform (RHOSP) uses OpenStack Object Storage (Swift) to store and serve user configuration files.

Swift is operated by a user account with the **swiftoperator** role.

Prerequisites

- A RHOSP administrator account on the target environment
- On Ceph RGW, the **account in url** option must be enabled

Procedure

To enable Swift on RHOSP:

1. As an administrator in the RHOSP CLI, add the **swiftoperator** role to the account that will access Swift:

```
$ openstack role add --user <user> --project <project> swiftoperator
```

-

Your RHOSP deployment can now use Swift to store and serve files.

1.1.4. Verifying external network access

The OpenShift Container Platform installer requires external network access. You must provide an external network value to it, or deployment fails. Before you run the installer, verify that a network with the External router type exists in Red Hat OpenStack Platform (RHOSP).

Prerequisites

- On RHOSP, the **NeutronDhcpAgentDnsmasqDnsServers** parameter must be configured to allow DHCP agents to forward instances' DNS queries. One way to set this parameter is to:

  a. Create a new environment file in the template directory.

  b. Provide parameter values in the file. For example:

  Sample neutron-dhcp-agent-dnsmasq-dns-servers.yaml file

  ```
  parameter_defaults:
    NeutronDhcpAgentDnsmasqDnsServers:
  ['<DNS_server_address_1>','<DNS_server_address_2']
  ```

  c. Include the environment file in your Overcloud deploy command. For example:

  ```
  $ openstack overcloud deploy --templates -e neutron-dhcp-agent-dnsmasq-dns-
  servers.yaml ...
  ```

Procedure

1. Using the RHOSP CLI, verify the name and ID of the 'External' network:

```
$ openstack network list --long -c ID -c Name -c "Router Type"

+--------------------------------------+----------------+-------------+
| ID                                   | Name           | Router Type |
+--------------------------------------+----------------+-------------+
| 148a8023-62a7-4672-b018-003462f8d7dc | public_network | External    |
+--------------------------------------+----------------+-------------+
```

A network with an External router type appears in the network list. If at least one does not, see Create an external network.

IMPORTANT

If the external network's CIDR range overlaps one of the default network ranges, you must change the matching network ranges in the **install-config.yaml** file before you run the installation program.

The default network ranges are:

Network	Range
machineCIDR	10.0.0.0/16
serviceNetwork	172.30.0.0/16
clusterNetwork	10.128.0.0/14

CAUTION

If the installation program finds multiple networks with the same name, it sets one of them at random. To avoid this behavior, create unique names for resources in RHOSP.

NOTE

If the Neutron trunk service plug-in is enabled, a trunk port is created by default. For more information, see Neutron trunk port.

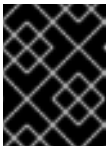
1.1.5. Defining parameters for the installation program

The OpenShift Container Platform installation program relies on a file called **clouds.yaml**. The file describes Red Hat OpenStack Platform (RHOSP) configuration parameters, including the project name, log in information, and authorization service URLs.

Procedure

1. Create the **clouds.yaml** file:

   - If your OpenStack distribution includes the Horizon web UI, generate a **clouds.yaml** file in it.

     IMPORTANT

     Remember to add a password to the **auth** field. You can also keep secrets in a separate file from **clouds.yaml**.

   - If your OpenStack distribution does not include the Horizon web UI, or you do not want to use Horizon, create the file yourself. For detailed information about **clouds.yaml**, see Config files in the RHOSP documentation.

     ```
     clouds:
       shiftstack:
         auth:
           auth_url: http://10.10.14.42:5000/v3
     ```

```
    project_name: shiftstack
    username: shiftstack_user
    password: XXX
    user_domain_name: Default
    project_domain_name: Default
dev-env:
  region_name: RegionOne
  auth:
    username: 'devuser'
    password: XXX
    project_name: 'devonly'
    auth_url: 'https://10.10.14.22:5001/v2.0'
```

2. Place the file that you generate in one of the following locations:

   a. The value of the **OS_CLIENT_CONFIG_FILE** environment variable

   b. The current directory

   c. A Unix-specific user configuration directory, for example **~/.config/openstack/clouds.yaml**

   d. A Unix-specific site configuration directory, for example **/etc/openstack/clouds.yaml**
   The installation program searches for **clouds.yaml** in that order.

1.1.6. Obtaining the installation program

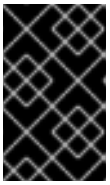
Before you install OpenShift Container Platform, download the installation file on a local computer.

Prerequisites

- You must install the cluster from a computer that uses Linux or macOS.
- You need 500 MB of local disk space to download the installation program.

Procedure

1. Access the Infrastructure Provider page on the Red Hat OpenShift Cluster Manager site. If you have a Red Hat account, log in with your credentials. If you do not, create an account.
2. Navigate to the page for your installation type, download the installation program for your operating system, and place the file in the directory where you will store the installation configuration files.



IMPORTANT

The installation program creates several files on the computer that you use to install your cluster. You must keep both the installation program and the files that the installation program creates after you finish installing the cluster.

3. Extract the installation program. For example, on a computer that uses a Linux operating system, run the following command:

```
$ tar xvf <installation_program>.tar.gz
```

4. From the Pull Secret page on the Red Hat OpenShift Cluster Manager site, download your installation pull secret as a **.txt** file. This pull secret allows you to authenticate with the services that are provided by the included authorities, including Quay.io, which serves the container images for OpenShift Container Platform components.

1.1.7. Creating the installation configuration file

You can customize your installation of OpenShift Container Platform on OpenStack.

Prerequisites

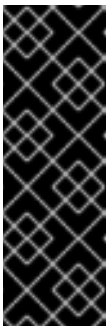
- Obtain the OpenShift Container Platform installation program and the pull secret for your cluster.

Procedure

1. Create the **install-config.yaml** file.
   a. Run the following command:

```
$ ./openshift-install create install-config --dir=<installation_directory> 1
```

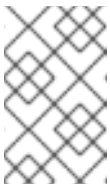
   1 For **<installation_directory>**, specify the directory name to store the files that the installation program creates.



IMPORTANT

Specify an empty directory. Some installation assets, like bootstrap X.509 certificates have short expiration intervals, so you must not reuse an installation directory. If you want to reuse individual files from another cluster installation, you can copy them into your directory. However, the file names for the installation assets might change between releases. Use caution when copying installation files from an earlier OpenShift Container Platform version.

   b. At the prompts, provide the configuration details for your cloud:
      i. Optional: Select an SSH key to use to access your cluster machines.



NOTE

For production OpenShift Container Platform clusters on which you want to perform installation debugging or disaster recovery on, specify an SSH key that your **ssh-agent** process uses.

      ii. Select **openstack** as the platform to target.
      iii. Specify the Red Hat OpenStack Platform (RHOSP) external network name to use for installing the cluster.
      iv. Specify the Floating IP address to use for external access to the OpenShift API.

v. Specify a RHOSP flavor with at least 16 GB RAM to use for control plane and compute nodes.

vi. Select the base domain to deploy the cluster to. All DNS records will be sub-domains of this base and will also include the cluster name.

vii. Enter a name for your cluster. The name must be 14 or fewer characters long.

viii. Paste the pull secret that you obtained from the Pull Secret page on the Red Hat OpenShift Cluster Manager site.

2. Modify the **install-config.yaml** file. You can find more information about the available parameters in the **Installation configuration parameters** section.

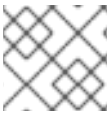
3. Back up the **install-config.yaml** file so that you can use it to install multiple clusters.

IMPORTANT

The **install-config.yaml** file is consumed during the installation process. If you want to reuse the file, you must back it up now.

1.1.8. Installation configuration parameters

Before you deploy an OpenShift Container Platform cluster, you provide parameter values to describe your account on the cloud platform that hosts your cluster and optionally customize your cluster's platform. When you create the **install-config.yaml** installation configuration file, you provide values for the required parameters through the command line. If you customize your cluster, you can modify the **install-config.yaml** file to provide more details about the platform.



NOTE

You cannot modify these parameters in the **install-config.yaml** file after installation.

Table 1.2. Required parameters

Parameter	Description	Values
baseDomain	The base domain of your cloud provider. This value is used to create routes to your OpenShift Container Platform cluster components. The full DNS name for your cluster is a combination of the **baseDomain** and **metadata.name** parameter values that uses the **<metadata.name>.<baseDomain>** format.	A fully-qualified domain or subdomain name, such as **example.com**.

Parameter	Description	Values
controlPlane.platform	The cloud provider to host the control plane machines. This parameter value must match the **compute.platform** parameter value.	**aws**, **azure**, **gcp**, **openstack**, or **{}**
compute.platform	The cloud provider to host the worker machines. This parameter value must match the **controlPlane.platform** parameter value.	**aws**, **azure**, **gcp**, **openstack**, or **{}**
metadata.name	The name of your cluster.	A string that contains uppercase or lowercase letters, such as **dev**. The string must be 14 characters or fewer long.
platform.<platform>.region	The region to deploy your cluster in.	A valid region for your cloud, such as **us-east-1** for AWS, **centralus** for Azure, or **region1** for Red Hat OpenStack Platform (RHOSP).

pullSecret	The pull secret that you obtained from the Pull Secret page on the Red Hat OpenShift Cluster Manager site. You use this pull secret to authenticate with the services that are provided by the included authorities, including Quay.io, which serves the container images for OpenShift Container Platform components.	{
		"auths":{
		"cloud.openshift.com":{
		"auth":"b3Blb=",
		"email":"you@example.com"
		},
		"quay.io":{
		"auth":"b3Blb=",
		"email":"you@example.com"
		}
		}
		}

Table 1.3. Optional parameters

Parameter	Description	Values

Parameter	Description	Values
sshKey	The SSH key to use to access your cluster machines.  **NOTE** For production OpenShift Container Platform clusters on which you want to perform installation debugging or disaster recovery on, specify an SSH key that your **ssh-agent** process uses.	A valid, local public SSH key that you added to the **ssh-agent** process.
fips	Whether to enable or disable FIPS mode. By default, FIPS mode is not enabled. If FIPS mode is enabled, the Red Hat Enterprise Linux CoreOS (RHCOS) machines that OpenShift Container Platform runs on bypass the default Kubernetes cryptography suite and use the cryptography modules that are provided with RHCOS instead.	**false** or **true**
publish	How to publish the user-facing endpoints of your cluster.	**Internal** or **External**. Set **publish** to **Internal** to deploy a private cluster, which cannot be accessed from the internet. The default value is **External**.

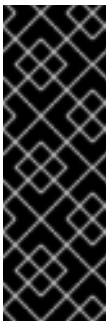
Parameter	Description	Values
compute.hyperthreading	Whether to enable or disable simultaneous multithreading, or **hyperthreading**, on compute machines. By default, simultaneous multithreading is enabled to increase the performance of your machines' cores.  **IMPORTANT** If you disable simultaneous multithreading, ensure that your capacity planning accounts for the dramatically decreased machine performance.	**Enabled** or **Disabled**
compute.replicas	The number of compute machines, which are also known as worker machines, to provision.	A positive integer greater than or equal to **2**. The default value is **3**.
controlPlane.hyperthreading	Whether to enable or disable simultaneous multithreading, or **hyperthreading**, on control plane machines. By default, simultaneous multithreading is enabled to increase the performance of your machines' cores.  **IMPORTANT** If you disable simultaneous multithreading, ensure that your capacity planning accounts for the dramatically decreased machine performance.	**Enabled** or **Disabled**
controlPlane.replicas	The number of control plane machines to provision.	A positive integer greater than or equal to **3**. The default value is **3**.

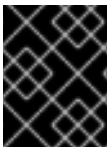
Table 1.4. Additional Red Hat OpenStack Platform (RHOSP) parameters

Parameter	Description	Values
compute.platform.openstack.rootVolume.size	For compute machines, the size in gigabytes of the root volume. If you do not set this value, machines use ephemeral storage.	Integer, for example **30**.
compute.platform.openstack.rootVolume.type	For compute machines, the root volume's type.	String, for example **performance**.
controlPlane.platform.openstack.rootVolume.size	For control plane machines, the size in gigabytes of the root volume. If you do not set this value, machines use ephemeral storage.	Integer, for example **30**.
controlPlane.platform.openstack.rootVolume.type	For control plane machines, the root volume's type.	String, for example **performance**.
platform.openstack.region	The region where the RHOSP cluster is created.	String, for example **region1**.
platform.openstack.cloud	The name of the RHOSP cloud to use from the list of clouds in the **clouds.yaml** file.	String, for example **MyCloud**.
platform.openstack.externalDNS	*Optional*. IP addresses for external DNS servers that cluster instances use for DNS resolution.	A list of IP addresses as strings, for example **["8.8.8.8", "192.168.1.12"]**.
platform.openstack.externalNetwork	The RHOSP external network name to be used for installation.	String, for example **external**.
platform.openstack.computeFlavor	The RHOSP flavor to use for control plane and compute machines.	String, for example **m1.xlarge**.
platform.openstack.lbFloatingIP	An existing floating IP address to associate with the load balancer API.	An IP address, for example **128.0.0.1**.

platform.openstack.defaultMachinePlatform	*Optional*. The default machine pool platform configuration.	<pre>{
		"type": "ml.large",
		"rootVolume": {
		"size": 30,
		"type": "performance"
		}
		}</pre>

1.1.8.1. Sample customized **install-config.yaml** file for OpenStack

This sample **install-config.yaml** demonstrates all of the possible Red Hat OpenStack Platform (RHOSP) customization options.



IMPORTANT

This sample file is provided for reference only. You must obtain your **install-config.yaml** file by using the installation program.

```
apiVersion: v1
baseDomain: example.com
clusterID: os-test
controlPlane:
  name: master
  platform: {}
  replicas: 3
compute:
- name: worker
  platform:
    openstack:
      type: ml.large
  replicas: 3
metadata:
  name: example
networking:
  clusterNetwork:
  - cidr: 10.128.0.0/14
    hostPrefix: 23
  machineCIDR: 10.0.0.0/16
  serviceNetwork:
  - 172.30.0.0/16
  networkType: OpenShiftSDN
platform:
  openstack:
    region: region1
    cloud: mycloud
    externalNetwork: external
    computeFlavor: m1.xlarge
    lbFloatingIP: 128.0.0.1
fips: false
pullSecret: '{"auths": ...}'
sshKey: ssh-ed25519 AAAA...
```

1.1.9. Generating an SSH private key and adding it to the agent

If you want to perform installation debugging or disaster recovery on your cluster, you must provide an SSH key to both your **ssh-agent** and to the installation program.

NOTE

In a production environment, you require disaster recovery and debugging.

You can use this key to SSH into the master nodes as the user **core**. When you deploy the cluster, the key is added to the **core** user's **~/.ssh/authorized_keys** list.

Procedure

1. If you do not have an SSH key that is configured for password-less authentication on your computer, create one. For example, on a computer that uses a Linux operating system, run the following command:

```
$ ssh-keygen -t rsa -b 4096 -N '' \
    -f <path>/<file_name> (1)
```

(1) Specify the path and file name, such as ~/**.ssh/id_rsa**, of the SSH key.

   Running this command generates an SSH key that does not require a password in the location that you specified.

2. Start the **ssh-agent** process as a background task:

```
$ eval "$(ssh-agent -s)"

Agent pid 31874
```

3. Add your SSH private key to the **ssh-agent**:

```
$ ssh-add <path>/<file_name> (1)

Identity added: /home/<you>/<path>/<file_name> (<computer_name>)
```

(1) Specify the path and file name for your SSH private key, such as ~/**.ssh/id_rsa**

Next steps

- When you install OpenShift Container Platform, provide the SSH public key to the installation program.

1.1.10. Enabling access to the environment

At deployment, all OpenShift Container Platform machines are created in a Red Hat OpenStack Platform (RHOSP)-tenant network. Therefore, they are not accessible directly in most RHOSP deployments.

You can configure the OpenShift Container Platform API to be accessible either with or without floating IP addresses.

1.1.10.1. Enabling access with floating IP addresses

Make OpenShift Container Platform API endpoints accessible by attaching two floating IP (FIP) addresses to them: one for the API load balancer (**lb FIP**), and one for OpenShift Container Platform applications (**apps FIP**).

IMPORTANT

The load balancer FIP is also used in the **install-config.yaml** file.

Procedure

1. Using the Red Hat OpenStack Platform (RHOSP) CLI, create a new external network:

```
$ openstack floating ip create <external network>
```

2. Add a record that follows this pattern to your DNS server:

```
api.<cluster name>.<base domain>  IN  A  <lb FIP>
```

NOTE

If you do not control the DNS server you can add the record to your **/etc/hosts** file instead. This action makes the API accessible to you only, which is not suitable for production deployment but does allow installation for development and testing.

TIP

You can make OpenShift Container Platform resources available outside of the cluster by assigning a floating IP address and updating your firewall configuration.

1.1.10.2. Enabling access without floating IP addresses

If you cannot use floating IP addresses, the OpenShift Container Platform installation might still finish. However, the installation program fails after it times out waiting for API access.

After the installation program times out, the cluster might still initialize. After the bootstrapping processing begins, it must complete. You must edit the cluster's networking configuration after it is deployed, however.

1.1.11. Deploy the cluster

You can install OpenShift Container Platform on a compatible cloud platform.

IMPORTANT

You can run the **create cluster** command of the installation program only once, during initial installation.

Prerequisites

- Have an administrator account on the target environment.
- Obtain the OpenShift Container Platform installation program and the pull secret for your cluster.

Procedure

1. Run the installation program:

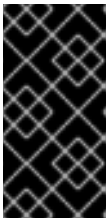
```
$ ./openshift-install create cluster --dir=<installation_directory> \ 1
    --log-level=info 2
```

1. For **<installation_directory>**, specify the location of your customized **./install-config.yaml** file.
2. To view different installation details, specify **warn**, **debug**, or **error** instead of **info**.

NOTE

If the cloud provider account that you configured on your host does not have sufficient permissions to deploy the cluster, the installation process stops, and the missing permissions are displayed.

When the cluster deployment completes, directions for accessing your cluster, including a link to its web console and credentials for the **kubeadmin** user, display in your terminal.



IMPORTANT

The Ignition config files that the installation program generates contain certificates that expire after 24 hours. You must keep the cluster running for 24 hours in a non-degraded state to ensure that the first certificate rotation has finished.

IMPORTANT

You must not delete the installation program or the files that the installation program creates. Both are required to delete the cluster.

1.1.12. Verifying cluster status

To verify your OpenShift Container Platform cluster's status during or after installation:

Procedure

1. In the cluster environment, export the administrator's kubeconfig file:

```
$ export KUBECONFIG=<installation_directory>/auth/kubeconfig 1
```

1. For **<installation_directory>**, specify the path to the directory that you stored the installation files in.

The **kubeconfig** file contains information about the cluster that is used by the CLI to connect a client to the correct cluster and API server.

2. View the control plane and compute machines created after a deployment:

```
$ oc get nodes
```

3. View your cluster's version:

```
$ oc get clusterversion
```

4. View your operators' status:

```
$ oc get clusteroperator
```

5. View all running Pods in the cluster:

```
$ oc get pods -A
```

1.1.13. Logging in to the cluster

You can log in to your cluster as a default system user by exporting the cluster **kubeconfig** file. The **kubeconfig** file contains information about the cluster that is used by the CLI to connect a client to the correct cluster and API server. The file is specific to a cluster and is created during OpenShift Container Platform installation.

Prerequisites

- Deploy an OpenShift Container Platform cluster.
- Install the **oc** CLI.

Procedure

1. Export the **kubeadmin** credentials:

```
$ export KUBECONFIG=<installation_directory>/auth/kubeconfig 1
```

1 For **<installation_directory>**, specify the path to the directory that you stored the installation files in.

2. Verify you can run **oc** commands successfully using the exported configuration:

```
$ oc whoami
system:admin
```

1.1.14. Configuring application access with floating IP addresses

After you install OpenShift Container Platform, configure Red Hat OpenStack Platform (RHOSP) to allow application network traffic.

Prerequisites

- OpenShift Container Platform cluster must be installed
- Floating IP addresses are enabled as described in *Enabling access to the environment*.

Procedure

After you install the OpenShift Container Platform cluster, attach a floating IP address to the ingress port:

1. Show the port:

```
$ openstack port show <cluster name>-<clusterID>-ingress-port
```

2. Attach the port to the IP address:

```
$ openstack floating ip set --port <ingress port ID> <apps FIP>
```

3. Add a wildcard **A** record for ***apps.** to your DNS file:

```
*.apps.<cluster name>.<base domain>  IN  A  <apps FIP>
```

NOTE

If you do not control the DNS server but want to enable application access for non-production purposes, you can add these hostnames to **/etc/hosts**:

```
<apps FIP> console-openshift-console.apps.<cluster name>.<base domain>
<apps FIP> integrated-oauth-server-openshift-authentication.apps.<cluster name>.
<base domain>
<apps FIP> oauth-openshift.apps.<cluster name>.<base domain>
<apps FIP> prometheus-k8s-openshift-monitoring.apps.<cluster name>.<base
domain>
<apps FIP> grafana-openshift-monitoring.apps.<cluster name>.<base domain>
<apps FIP> <app name>.apps.<cluster name>.<base domain>
```

Next steps

- Customize your cluster.
- If necessary, you can opt out of remote health reporting .

1.2. INSTALLING A CLUSTER ON OPENSTACK WITH KURYR

In OpenShift Container Platform version 4.3, you can install a customized cluster on Red Hat OpenStack Platform (RHOSP) that uses Kuryr SDN. To customize the installation, modify parameters in the **install-config.yaml** before you install the cluster.

Prerequisites

- Review details about the OpenShift Container Platform installation and update processes.
- Have access to an RHOSP administrator's account

1.2.1. About Kuryr SDN

Kuryr is a container network interface (CNI) plug-in solution that uses OpenStack Neutron and OpenStack Octavia to provide networking for Pods and Services.

Kuryr and OpenShift Container Platform integration is primarily designed for OpenShift Container Platform clusters running on OpenStack VMs. Kuryr improves the network performance by plugging OpenShift Pods into OpenStack SDN. In addition, it provides interconnectivity between OpenShift Pods and OpenStack virtual instances.

Kuryr components are installed as Pods in OpenShift Container Platform using the **openshift-kuryr** namespace:

- **kuryr-controller** - a single Service instance installed on a **master** node. This is modeled in OpenShift Container Platform as a **Deployment**.
- **kuryr-cni** - a container installing and configuring Kuryr as a CNI driver on each OpenShift Container Platform node. This is modeled in OpenShift Container Platform as a **DaemonSet**.

The Kuryr controller watches the OpenShift API server for Pod, Service, and namespace create, update, and delete events. It maps the OpenShift Container Platform API calls to corresponding objects in Neutron and Octavia. This means that every network solution that implements the Neutron trunk port functionality can be used to back OpenShift Container Platform via Kuryr. This includes open source solutions such as Open vSwitch (OVS) and Open Virtual Network (OVN) as well as Neutron-compatible commercial SDNs.

Kuryr is recommended for OpenShift deployments on encapsulated OpenStack tenant networks to avoid double encapsulation, such as running an encapsulated OpenShift SDN over an OpenStack network.

Conversely, Kuryr is not recommended in the following cases:

- You use provider networks or tenant VLANs.
- Your deployment uses many Services on a few hypervisors. Each OpenShift Service creates an Octavia Amphora virtual machine in OpenStack that hosts a required load balancer.
- UDP Services are needed.

1.2.2. Resource guidelines for installing OpenShift Container Platform on OpenStack with Kuryr

When using Kuryr SDN, the Pods, Services, namespaces, and network policies are using resources from the RHOSP quota; this increases the minimum requirements. Kuryr also has some additional requirements on top of what a default install requires.

Use the following quota to satisfy a default cluster's minimum requirements:

Table 1.5. Recommended resources for a default OpenShift Container Platform cluster on RHOSP with Kuryr

Resource	Value
Floating IP addresses	3 - plus the expected number of Services of LoadBalancer type
Ports	1500 - 1 needed per Pod
Routers	1
Subnets	250 - 1 needed per Namespace/Project
Networks	250 - 1 needed per Namespace/Project
RAM	112 GB

Resource	Value
vCPUs	28
Volume storage	175 GB
Instances	7
Security groups	250 - 1 needed per Service and per NetworkPolicy
Security group rules	1000
Swift containers	2
Swift objects	1
Swift available space	10 MB or more
Load balancers	100 - 1 needed per Service
Load balancer listeners	500 - 1 needed per Service-exposed port
Load balancer pools	500 - 1 needed per Service-exposed port

A cluster might function with fewer than recommended resources, but its performance is not guaranteed.

Take the following notes into consideration when setting resources:

- The number of ports required is actually larger than the number of Pods. Kuryr uses ports pools to have pre-created ports ready to be used by Pods and speed up the Pods booting time.
- Each NetworkPolicy is mapped into an RHOSP security group, and depending on the NetworkPolicy spec, one or more rules are added to the security group.
- Each Service is mapped into an RHOSP load balancer. Each load balancer has a security group with the user project; therefore, it must be taken into account when estimating the number of security groups required for the quota.
- Swift space requirements vary depending on the size of the bootstrap Ignition file and image registry.
- The quota does not account for load balancer resources (such as VM resources), but you must consider these resources when you decide the RHOSP deployment's size. The default installation will have more than 50 load balancers; the clusters must be able to accommodate them.

An OpenShift Container Platform deployment comprises control plane machines, compute machines, and a bootstrap machine.

To enable Kuryr SDN, your environment must meet the following requirements:

- Run OpenStack 13+.
- Have Overcloud with Octavia.
- Use Neutron Trunk ports extension.
- Use **openvswitch** firewall driver if ML2/OVS Neutron driver is used instead of **ovs-hybrid**.

1.2.2.1. Increasing quota

When using Kuryr SDN, you must increase quotas to satisfy the OpenStack resources used by Pods, Services, namespaces, and network policies.

Procedure

- Increase the quotas for a project by running the following command:

```
$ sudo openstack quota set --secgroups 250 --secgroup-rules 1000 --ports 1500 --subnets
250 --networks 250 <project>
```

1.2.2.2. Configuring Neutron

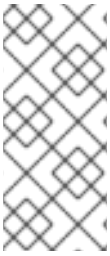
Kuryr CNI leverages the Neutron Trunks extension to plug containers into the OpenStack SDN, so you must use the **trunks** extension for Kuryr to properly work.

In addition, if you leverage the default ML2/OVS Neutron driver, the firewall must be set to **openvswitch** instead of **ovs_hybrid** so that security groups are enforced on trunk subports and Kuryr can properly handle network policies.

1.2.2.3. Configuring Octavia

Kuryr SDN uses OpenStack Octavia LBaaS to implement OpenShift Services. Thus, you must install and configure Octavia components in your OpenStack environment to use Kuryr SDN.

To enable Octavia, you must include the Octavia Service during the installation of the OpenStack Overcloud, or upgrade the Octavia Service if the Overcloud already exists. The following steps for enabling Octavia apply to both a clean install of the Overcloud or an Overcloud update.



NOTE

The following steps only capture the key pieces required during the deployment of OpenStack when dealing with Octavia. It is also important to note that registry methods vary.

This example uses the local registry method.

Procedure

1. If you are using the local registry, create a template to upload the images to the registry. For example:

```
(undercloud) $ openstack overcloud container image prepare \
-e /usr/share/openstack-tripleo-heat-templates/environments/services-docker/octavia.yaml \
--namespace=registry.access.redhat.com/rhosp13 \
```

```
--push-destination=<local-ip-from-undercloud.conf>:8787 \
--prefix=openstack- \
--tag-from-label {version}-{release} \
--output-env-file=/home/stack/templates/overcloud_images.yaml \
--output-images-file /home/stack/local_registry_images.yaml
```

2. Verify that the **local_registry_images.yaml** file contains the Octavia images. For example:

```
...
- imagename: registry.access.redhat.com/rhosp13/openstack-octavia-api:13.0-43
  push_destination: <local-ip-from-undercloud.conf>:8787
- imagename: registry.access.redhat.com/rhosp13/openstack-octavia-health-manager:13.0-
45
  push_destination: <local-ip-from-undercloud.conf>:8787
- imagename: registry.access.redhat.com/rhosp13/openstack-octavia-housekeeping:13.0-45
  push_destination: <local-ip-from-undercloud.conf>:8787
- imagename: registry.access.redhat.com/rhosp13/openstack-octavia-worker:13.0-44
  push_destination: <local-ip-from-undercloud.conf>:8787
```

NOTE

The Octavia container versions vary depending upon the specific RHOSP release installed.

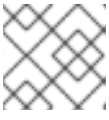
3. Pull the container images from registry.redhat.io to the Undercloud node:

```
(undercloud) $ sudo openstack overcloud container image upload \
  --config-file  /home/stack/local_registry_images.yaml \
  --verbose
```

This may take some time depending on the speed of your network and Undercloud disk.

4. Since an Octavia load balancer is used to access the OpenShift API, you must increase their listeners' default timeouts for the connections. The default timeout is 50 seconds. Increase the timeout to 20 minutes by passing the following file to the Overcloud deploy command:

```
(undercloud) $ cat octavia_timeouts.yaml
parameter_defaults:
  OctaviaTimeoutClientData: 1200000
  OctaviaTimeoutMemberData: 1200000
```



NOTE

This is not needed for Red Hat OpenStack Platform 14+.

5. Install or update your Overcloud environment with Octavia:

```
openstack overcloud deploy --templates \
  -e /usr/share/openstack-tripleo-heat-templates/environments/services-docker/octavia.yaml \
  -e octavia_timeouts.yaml
```

NOTE

This command only includes the files associated with Octavia; it varies based on your specific installation of OpenStack. See the official OpenStack documentation for further information. For more information on customizing your Octavia installation, see installation of Octavia using Director.

NOTE

When leveraging Kuryr SDN, the Overcloud installation requires the Neutron **trunk** extension. This is available by default on Director deployments. Use the **openvswitch** firewall instead of the default **ovs-hybrid** when the Neutron backend is ML2/OVS. There is no need for modifications if the backend is ML2/OVN.

6. To enforce network policies across Services, like when traffic goes through the Octavia load balancer, you must ensure Octavia creates the Amphora VM security groups on the user project. To do that, you must add the project ID to the **octavia.conf** configuration file after you create the project.
This ensures that required LoadBalancer security groups belong to that project and that they can be updated to enforce Services isolation.

   a. Get the project ID

```
$ openstack project show <project>
+-------------+----------------------------------+
| Field       | Value                            |
+-------------+----------------------------------+
| description |                                  |
| domain_id   | default                          |
| enabled     | True                             |
| id          | PROJECT_ID                       |
| is_domain   | False                            |
| name        | *<project>*                      |
| parent_id   | default                          |
| tags        | []                               |
+-------------+----------------------------------+
```

   b. Add the project ID to **octavia.conf** for the controllers.

      i. List the Overcloud controllers.

```
$ source stackrc  # Undercloud credentials
$ openstack server list
+--------------------------------------+--------------+--------+----------------------+------------
----+------------+
|
| ID                                   | Name         | Status | Networks
| Image          | Flavor     |
|
+--------------------------------------+--------------+--------+----------------------+------------
----+------------+
|
| 6bef8e73-2ba5-4860-a0b1-3937f8ca7e01 | controller-0 | ACTIVE |
ctlplane=192.168.24.8 | overcloud-full | controller |
```

```
|
| dda3173a-ab26-47f8-a2dc-8473b4a67ab9 | compute-0    | ACTIVE |
ctlplane=192.168.24.6 | overcloud-full | compute    |
|
+--------------------------------------+-------------+--------+----------------------+------------
----+------------+
```

ii. SSH into the controller(s).

```
$ ssh heat-admin@192.168.24.8
```

iii. Edit the **octavia.conf** to add the project into the list of projects where Amphora security groups are on the user's account.

```
# List of project IDs that are allowed to have Load balancer security groups
# belonging to them.
amp_secgroup_allowed_projects = PROJECT_ID
```

c. Restart the Octavia worker so the new configuration loads.

```
controller-0$ sudo docker restart octavia_worker
```

NOTE

Depending on your OpenStack environment, Octavia might not support UDP listeners, which means there is no support for UDP Services if Kuryr SDN is used.

1.2.2.4. Known limitations of installing with Kuryr

There are known limitations when using Kuryr SDN:

- An Amphora load balancer VM is deployed per OpenShift Service with the default Octavia load balancer driver (Amphora driver). If the environment is resource constrained, creating a large amount of Services could be a problem.
- Depending on the Octavia version, UDP listeners are not supported. This means that OpenShift UDP Services are not supported.
- There is a known limitation of Octavia not supporting listeners on different protocols, like UDP and TCP, on the same port. Thus, Services exposing the same port for different protocols are not supported.
- Due to the above UDP limitations of Octavia, Kuryr forces Pods to use TCP for DNS resolution. This is set with the **use-vc** option in **resolv.conf**. This might be a problem for Pods running Go applications compiled with the **CGO_ENABLED** flag disabled, as that uses the **go** resolver that only leverages UDP and is not considering the **use-vc** option added by Kuryr to the **resolv.conf**. This is a problem also for musl-based containers as its resolver does not support the **use-vc** option. This includes images built from **alpine**.

1.2.2.5. Control plane and compute machines

By default, the OpenShift Container Platform installation program stands up three control plane and compute machines.

Each machine requires:

- An instance from the RHOSP quota
- A port from the RHOSP quota
- A flavor with at least 16 GB memory, 4 vCPUs, and 25 GB storage space

TIP

Compute machines host the applications that you run on OpenShift Container Platform; aim to run as many as you can.

1.2.2.6. Bootstrap machine

During installation, a bootstrap machine is temporarily provisioned to stand up the control plane. After the production control plane is ready, the bootstrap machine is deprovisioned.

The bootstrap machine requires:

- An instance from the RHOSP quota
- A port from the RHOSP quota
- A flavor with at least 16 GB memory, 4 vCPUs, and 25 GB storage space

NOTE

The installation program cannot pass certificate authority bundles to Ignition on control plane machines. Therefore, the bootstrap machine cannot retrieve Ignition configurations from Swift if your endpoint uses self-signed certificates.

1.2.3. Internet and Telemetry access for OpenShift Container Platform

In OpenShift Container Platform 4.3, you require access to the internet to install and entitle your cluster. The Telemetry service, which runs by default to provide metrics about cluster health and the success of updates, also requires internet access. If your cluster is connected to the internet, Telemetry runs automatically, and your cluster is registered to the Red Hat OpenShift Cluster Manager . From there, you can allocate entitlements to your cluster.

You must have internet access to:

- Access the Red Hat OpenShift Cluster Manager page to download the installation program and perform subscription management and entitlement. If the cluster has internet access and you do not disable Telemetry, that service automatically entitles your cluster. If the Telemetry service cannot entitle your cluster, you must manually entitle it on the Cluster registration page.
- Access Quay.io to obtain the packages that are required to install your cluster.
- Obtain the packages that are required to perform cluster updates.

IMPORTANT

If your cluster cannot have direct internet access, you can perform a restricted network installation on some types of infrastructure that you provision. During that process, you download the content that is required and use it to populate a mirror registry with the packages that you need to install a cluster and generate the installation program. With some installation types, the environment that you install your cluster in will not require internet access. Before you update the cluster, you update the content of the mirror registry.

1.2.4. Enabling Swift on OpenStack

OpenShift Container Platform on Red Hat OpenStack Platform (RHOSP) uses OpenStack Object Storage (Swift) to store and serve user configuration files.

Swift is operated by a user account with the **swiftoperator** role.

Prerequisites

- A RHOSP administrator account on the target environment
- On Ceph RGW, the **account in url** option must be enabled

Procedure

To enable Swift on RHOSP:

1. As an administrator in the RHOSP CLI, add the **swiftoperator** role to the account that will access Swift:

```
$ openstack role add --user <user> --project <project> swiftoperator
```

Your RHOSP deployment can now use Swift to store and serve files.

1.2.5. Verifying external network access

The OpenShift Container Platform installer requires external network access. You must provide an external network value to it, or deployment fails. Before you run the installer, verify that a network with the External router type exists in Red Hat OpenStack Platform (RHOSP).

Prerequisites

- On RHOSP, the **NeutronDhcpAgentDnsmasqDnsServers** parameter must be configured to allow DHCP agents to forward instances' DNS queries. One way to set this parameter is to:
  a. Create a new environment file in the template directory.
  b. Provide parameter values in the file. For example:

Sample neutron-dhcp-agent-dnsmasq-dns-servers.yaml file

```
parameter_defaults:
  NeutronDhcpAgentDnsmasqDnsServers:
['<DNS_server_address_1>','<DNS_server_address_2']
```

c. Include the environment file in your Overcloud deploy command. For example:

```
$ openstack overcloud deploy --templates -e neutron-dhcp-agent-dnsmasq-dns-servers.yaml ...
```

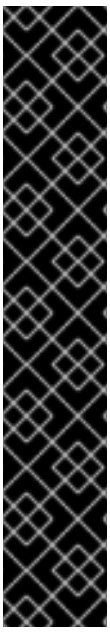
Procedure

1. Using the RHOSP CLI, verify the name and ID of the 'External' network:

```
$ openstack network list --long -c ID -c Name -c "Router Type"

+--------------------------------------+----------------+-------------+
| ID                                   | Name           | Router Type |
+--------------------------------------+----------------+-------------+
| 148a8023-62a7-4672-b018-003462f8d7dc | public_network | External    |
+--------------------------------------+----------------+-------------+
```

A network with an External router type appears in the network list. If at least one does not, see Create an external network.



IMPORTANT

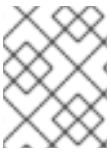
If the external network's CIDR range overlaps one of the default network ranges, you must change the matching network ranges in the **install-config.yaml** file before you run the installation program.

The default network ranges are:

Network	Range
machineCIDR	10.0.0.0/16
serviceNetwork	172.30.0.0/16
clusterNetwork	10.128.0.0/14

CAUTION

If the installation program finds multiple networks with the same name, it sets one of them at random. To avoid this behavior, create unique names for resources in RHOSP.



NOTE

If the Neutron trunk service plug-in is enabled, a trunk port is created by default. For more information, see Neutron trunk port .

1.2.6. Defining parameters for the installation program

The OpenShift Container Platform installation program relies on a file called **clouds.yaml**. The file describes Red Hat OpenStack Platform (RHOSP) configuration parameters, including the project name, log in information, and authorization service URLs.

Procedure

1. Create the **clouds.yaml** file:

   - If your OpenStack distribution includes the Horizon web UI, generate a **clouds.yaml** file in it.

     IMPORTANT

     Remember to add a password to the **auth** field. You can also keep secrets in a separate file from **clouds.yaml**.

   - If your OpenStack distribution does not include the Horizon web UI, or you do not want to use Horizon, create the file yourself. For detailed information about **clouds.yaml**, see Config files in the RHOSP documentation.

     ```
     clouds:
       shiftstack:
         auth:
           auth_url: http://10.10.14.42:5000/v3
           project_name: shiftstack
           username: shiftstack_user
           password: XXX
           user_domain_name: Default
           project_domain_name: Default
       dev-env:
         region_name: RegionOne
         auth:
           username: 'devuser'
           password: XXX
           project_name: 'devonly'
           auth_url: 'https://10.10.14.22:5001/v2.0'
     ```

2. Place the file that you generate in one of the following locations:

   a. The value of the **OS_CLIENT_CONFIG_FILE** environment variable

   b. The current directory

   c. A Unix-specific user configuration directory, for example ~/**.config/openstack/clouds.yaml**

   d. A Unix-specific site configuration directory, for example /**etc/openstack/clouds.yaml**
   The installation program searches for **clouds.yaml** in that order.

1.2.7. Obtaining the installation program

Before you install OpenShift Container Platform, download the installation file on a local computer.

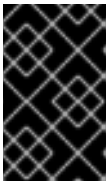
Prerequisites

- You must install the cluster from a computer that uses Linux or macOS.
- You need 500 MB of local disk space to download the installation program.

Procedure

1. Access the Infrastructure Provider page on the Red Hat OpenShift Cluster Manager site. If you have a Red Hat account, log in with your credentials. If you do not, create an account.

2. Navigate to the page for your installation type, download the installation program for your operating system, and place the file in the directory where you will store the installation configuration files.



IMPORTANT

The installation program creates several files on the computer that you use to install your cluster. You must keep both the installation program and the files that the installation program creates after you finish installing the cluster.

3. Extract the installation program. For example, on a computer that uses a Linux operating system, run the following command:

```
$ tar xvf <installation_program>.tar.gz
```

4. From the Pull Secret page on the Red Hat OpenShift Cluster Manager site, download your installation pull secret as a **.txt** file. This pull secret allows you to authenticate with the services that are provided by the included authorities, including Quay.io, which serves the container images for OpenShift Container Platform components.

1.2.8. Creating the installation configuration file

You can customize your installation of OpenShift Container Platform on OpenStack.

Prerequisites

- Obtain the OpenShift Container Platform installation program and the pull secret for your cluster.

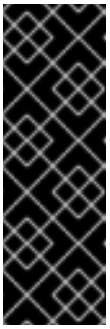
Procedure

1. Create the **install-config.yaml** file.

   a. Run the following command:

```
$ ./openshift-install create install-config --dir=<installation_directory> 1
```

1 For **<installation_directory>**, specify the directory name to store the files that the installation program creates.



IMPORTANT

Specify an empty directory. Some installation assets, like bootstrap X.509 certificates have short expiration intervals, so you must not reuse an installation directory. If you want to reuse individual files from another cluster installation, you can copy them into your directory. However, the file names for the installation assets might change between releases. Use caution when copying installation files from an earlier OpenShift Container Platform version.

b. At the prompts, provide the configuration details for your cloud:

i. Optional: Select an SSH key to use to access your cluster machines.

NOTE

For production OpenShift Container Platform clusters on which you want to perform installation debugging or disaster recovery on, specify an SSH key that your **ssh-agent** process uses.

ii. Select **openstack** as the platform to target.

iii. Specify the Red Hat OpenStack Platform (RHOSP) external network name to use for installing the cluster.

iv. Specify the Floating IP address to use for external access to the OpenShift API.

v. Specify a RHOSP flavor with at least 16 GB RAM to use for control plane and compute nodes.

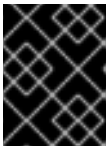
vi. Select the base domain to deploy the cluster to. All DNS records will be sub-domains of this base and will also include the cluster name.

vii. Enter a name for your cluster. The name must be 14 or fewer characters long.

viii. Paste the pull secret that you obtained from the Pull Secret page on the Red Hat OpenShift Cluster Manager site.

2. Modify the **install-config.yaml** file. You can find more information about the available parameters in the **Installation configuration parameters** section.

3. Back up the **install-config.yaml** file so that you can use it to install multiple clusters.



IMPORTANT

The **install-config.yaml** file is consumed during the installation process. If you want to reuse the file, you must back it up now.

1.2.9. Installation configuration parameters

Before you deploy an OpenShift Container Platform cluster, you provide parameter values to describe your account on the cloud platform that hosts your cluster and optionally customize your cluster's platform. When you create the **install-config.yaml** installation configuration file, you provide values for the required parameters through the command line. If you customize your cluster, you can modify the **install-config.yaml** file to provide more details about the platform.

NOTE

You cannot modify these parameters in the **install-config.yaml** file after installation.

Table 1.6. Required parameters

Parameter	Description	Values
baseDomain	The base domain of your cloud provider. This value is used to create routes to your OpenShift Container Platform cluster components. The full DNS name for your cluster is a combination of the **baseDomain** and **metadata.name** parameter values that uses the **<metadata.name>.<baseDomain>** format.	A fully-qualified domain or subdomain name, such as **example.com**.
controlPlane.platform	The cloud provider to host the control plane machines. This parameter value must match the **compute.platform** parameter value.	**aws**, **azure**, **gcp**, **openstack**, or **{}**
compute.platform	The cloud provider to host the worker machines. This parameter value must match the **controlPlane.platform** parameter value.	**aws**, **azure**, **gcp**, **openstack**, or **{}**
metadata.name	The name of your cluster.	A string that contains uppercase or lowercase letters, such as **dev**. The string must be 14 characters or fewer long.
platform.<platform>.region	The region to deploy your cluster in.	A valid region for your cloud, such as **us-east-1** for AWS, **centralus** for Azure, or **region1** for Red Hat OpenStack Platform (RHOSP).

pullSecret	The pull secret that you obtained from the Pull Secret page on the Red Hat OpenShift Cluster Manager site. You use this pull secret to authenticate with the services that are provided by the included authorities, including Quay.io, which serves the container images for OpenShift Container Platform components.	{
		"auths":{
		"cloud.openshift.com":{
		"auth":"b3Blb=",
		"email":"you@example.com"
		},
		"quay.io":{
		"auth":"b3Blb=",
		"email":"you@example.com"
		}
		}
		}

Table 1.7. Optional parameters

Parameter	Description	Values
sshKey	The SSH key to use to access your cluster machines.  **NOTE** For production OpenShift Container Platform clusters on which you want to perform installation debugging or disaster recovery on, specify an SSH key that your **ssh-agent** process uses.	A valid, local public SSH key that you added to the **ssh-agent** process.
fips	Whether to enable or disable FIPS mode. By default, FIPS mode is not enabled. If FIPS mode is enabled, the Red Hat Enterprise Linux CoreOS (RHCOS) machines that OpenShift Container Platform runs on bypass the default Kubernetes cryptography suite and use the cryptography modules that are provided with RHCOS instead.	**false** or **true**
publish	How to publish the user-facing endpoints of your cluster.	**Internal** or **External**. Set **publish** to **Internal** to deploy a private cluster, which cannot be accessed from the internet. The default value is **External**.
compute.hyperthreading	Whether to enable or disable simultaneous multithreading, or **hyperthreading**, on compute machines. By default, simultaneous multithreading is enabled to increase the performance of your machines' cores.  **IMPORTANT** If you disable simultaneous multithreading, ensure that your capacity planning accounts for the dramatically decreased machine performance.	**Enabled** or **Disabled**

Parameter	Description	Values
compute.replicas	The number of compute machines, which are also known as worker machines, to provision.	A positive integer greater than or equal to **2**. The default value is **3**.
controlPlane.hyperthreading	Whether to enable or disable simultaneous multithreading, or **hyperthreading**, on control plane machines. By default, simultaneous multithreading is enabled to increase the performance of your machines' cores.  **IMPORTANT** If you disable simultaneous multithreading, ensure that your capacity planning accounts for the dramatically decreased machine performance.	**Enabled** or **Disabled**
controlPlane.replicas	The number of control plane machines to provision.	A positive integer greater than or equal to **3**. The default value is **3**.

Table 1.8. Additional Red Hat OpenStack Platform (RHOSP) parameters

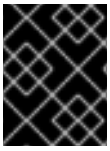
Parameter	Description	Values
compute.platform.openstack.rootVolume.size	For compute machines, the size in gigabytes of the root volume. If you do not set this value, machines use ephemeral storage.	Integer, for example **30**.
compute.platform.openstack.rootVolume.type	For compute machines, the root volume's type.	String, for example **performance**.
controlPlane.platform.openstack.rootVolume.size	For control plane machines, the size in gigabytes of the root volume. If you do not set this value, machines use ephemeral storage.	Integer, for example **30**.
controlPlane.platform.openstack.rootVolume.type	For control plane machines, the root volume's type.	String, for example **performance**.

Parameter	Description	Values
platform.openstack.region	The region where the RHOSP cluster is created.	String, for example **region1**.
platform.openstack.cloud	The name of the RHOSP cloud to use from the list of clouds in the **clouds.yaml** file.	String, for example **MyCloud**.
platform.openstack.externalDNS	*Optional*. IP addresses for external DNS servers that cluster instances use for DNS resolution.	A list of IP addresses as strings, for example **["8.8.8.8", "192.168.1.12"]**.
platform.openstack.externalNetwork	The RHOSP external network name to be used for installation.	String, for example **external**.
platform.openstack.computeFlavor	The RHOSP flavor to use for control plane and compute machines.	String, for example **m1.xlarge**.
platform.openstack.lbFloatingIP	An existing floating IP address to associate with the load balancer API.	An IP address, for example **128.0.0.1**.

platform.openstack.defaultMachinePlatform	*Optional*. The default machine pool platform configuration.	<pre>{
		"type": "ml.large",
		"rootVolume": {
		"size": 30,
		"type": "performance"
		}
		}</pre>

1.2.9.1. Sample customized **install-config.yaml** file for OpenStack with Kuryr

To deploy with Kuryr SDN instead of the default OpenShift SDN, you must modify the **install-config.yaml** file to include **Kuryr** as the desired **networking.networkType** and proceed with the default OpenShift SDN installation steps. This sample **install-config.yaml** demonstrates all of the possible Red Hat OpenStack Platform (RHOSP) customization options.



IMPORTANT

This sample file is provided for reference only. You must obtain your **install-config.yaml** file by using the installation program.

```
apiVersion: v1
baseDomain: example.com
clusterID: os-test
controlPlane:
  name: master
```

```
  platform: {}
  replicas: 3
compute:
- name: worker
  platform:
    openstack:
      type: ml.large
  replicas: 3
metadata:
  name: example
networking:
  clusterNetwork:
  - cidr: 10.128.0.0/14
    hostPrefix: 23
  machineCIDR: 10.0.0.0/16
  serviceNetwork:
  - 172.30.0.0/16
  networkType: Kuryr
platform:
  openstack:
    region: region1
    cloud: mycloud
    externalNetwork: external
    computeFlavor: m1.xlarge
    lbFloatingIP: 128.0.0.1
    trunkSupport: true
    octaviaSupport: true
pullSecret: '{"auths": ...}'
sshKey: ssh-ed25519 AAAA...
```

NOTE

Both **trunkSupport** and **octaviaSupport** are automatically discovered by the installer, so there is no need to set them. But if your environment does not meet both requirements, Kuryr SDN will not properly work. Trunks are needed to connect the Pods to the OpenStack network and Octavia is required to create the OpenShift Services.

1.2.10. Generating an SSH private key and adding it to the agent

If you want to perform installation debugging or disaster recovery on your cluster, you must provide an SSH key to both your **ssh-agent** and to the installation program.

NOTE

In a production environment, you require disaster recovery and debugging.

You can use this key to SSH into the master nodes as the user **core**. When you deploy the cluster, the key is added to the **core** user's **~/.ssh/authorized_keys** list.

Procedure

1. If you do not have an SSH key that is configured for password-less authentication on your computer, create one. For example, on a computer that uses a Linux operating system, run the following command:

```
$ ssh-keygen -t rsa -b 4096 -N '' \
    -f <path>/<file_name> 1
```

1 Specify the path and file name, such as ~/**.ssh/id_rsa**, of the SSH key.

Running this command generates an SSH key that does not require a password in the location that you specified.

2. Start the **ssh-agent** process as a background task:

```
$ eval "$(ssh-agent -s)"

Agent pid 31874
```

3. Add your SSH private key to the **ssh-agent**:

```
$ ssh-add <path>/<file_name> 1

Identity added: /home/<you>/<path>/<file_name> (<computer_name>)
```

1 Specify the path and file name for your SSH private key, such as ~/**.ssh/id_rsa**

Next steps

- When you install OpenShift Container Platform, provide the SSH public key to the installation program.

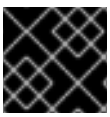
1.2.11. Enabling access to the environment

At deployment, all OpenShift Container Platform machines are created in a Red Hat OpenStack Platform (RHOSP)-tenant network. Therefore, they are not accessible directly in most RHOSP deployments.

You can configure the OpenShift Container Platform API to be accessible either with or without floating IP addresses.

1.2.11.1. Enabling access with floating IP addresses

Make OpenShift Container Platform API endpoints accessible by attaching two floating IP (FIP) addresses to them: one for the API load balancer (**lb FIP**), and one for OpenShift Container Platform applications (**apps FIP**).



IMPORTANT

The load balancer FIP is also used in the **install-config.yaml** file.

Procedure

1. Using the Red Hat OpenStack Platform (RHOSP) CLI, create a new external network:

```
$ openstack floating ip create <external network>
```

2. Add a record that follows this pattern to your DNS server:

```
api.<cluster name>.<base domain>  IN  A  <lb FIP>
```

NOTE

If you do not control the DNS server you can add the record to your **/etc/hosts** file instead. This action makes the API accessible to you only, which is not suitable for production deployment but does allow installation for development and testing.

TIP

You can make OpenShift Container Platform resources available outside of the cluster by assigning a floating IP address and updating your firewall configuration.

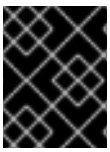
1.2.11.2. Enabling access without floating IP addresses

If you cannot use floating IP addresses, the OpenShift Container Platform installation might still finish. However, the installation program fails after it times out waiting for API access.

After the installation program times out, the cluster might still initialize. After the bootstrapping processing begins, it must complete. You must edit the cluster's networking configuration after it is deployed, however.

1.2.12. Deploy the cluster

You can install OpenShift Container Platform on a compatible cloud platform.



IMPORTANT

You can run the **create cluster** command of the installation program only once, during initial installation.

Prerequisites

- Have an administrator account on the target environment.
- Obtain the OpenShift Container Platform installation program and the pull secret for your cluster.

Procedure

1. Run the installation program:

```
$ ./openshift-install create cluster --dir=<installation_directory> \ 1
    --log-level=info 2
```

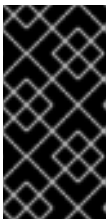
1 For **<installation_directory>**, specify the location of your customized **./install-config.yaml** file.

2 To view different installation details, specify **warn**, **debug**, or **error** instead of **info**.

NOTE

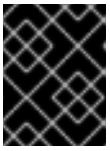
If the cloud provider account that you configured on your host does not have sufficient permissions to deploy the cluster, the installation process stops, and the missing permissions are displayed.

When the cluster deployment completes, directions for accessing your cluster, including a link to its web console and credentials for the **kubeadmin** user, display in your terminal.



IMPORTANT

The Ignition config files that the installation program generates contain certificates that expire after 24 hours. You must keep the cluster running for 24 hours in a non-degraded state to ensure that the first certificate rotation has finished.



IMPORTANT

You must not delete the installation program or the files that the installation program creates. Both are required to delete the cluster.

1.2.13. Verifying cluster status

To verify your OpenShift Container Platform cluster's status during or after installation:

Procedure

1. In the cluster environment, export the administrator's kubeconfig file:

```
$ export KUBECONFIG=<installation_directory>/auth/kubeconfig 1
```

1 For **<installation_directory>**, specify the path to the directory that you stored the installation files in.

The **kubeconfig** file contains information about the cluster that is used by the CLI to connect a client to the correct cluster and API server.

2. View the control plane and compute machines created after a deployment:

```
$ oc get nodes
```

3. View your cluster's version:

```
$ oc get clusterversion
```

4. View your operators' status:

```
$ oc get clusteroperator
```

5. View all running Pods in the cluster:

```
$ oc get pods -A
```

1.2.14. Logging in to the cluster

You can log in to your cluster as a default system user by exporting the cluster **kubeconfig** file. The **kubeconfig** file contains information about the cluster that is used by the CLI to connect a client to the correct cluster and API server. The file is specific to a cluster and is created during OpenShift Container Platform installation.

Prerequisites

- Deploy an OpenShift Container Platform cluster.
- Install the **oc** CLI.

Procedure

1. Export the **kubeadmin** credentials:

```
$ export KUBECONFIG=<installation_directory>/auth/kubeconfig 1
```

1 For **<installation_directory>**, specify the path to the directory that you stored the installation files in.

2. Verify you can run **oc** commands successfully using the exported configuration:

```
$ oc whoami
system:admin
```

1.2.15. Configuring application access with floating IP addresses

After you install OpenShift Container Platform, configure Red Hat OpenStack Platform (RHOSP) to allow application network traffic.

Prerequisites

- OpenShift Container Platform cluster must be installed
- Floating IP addresses are enabled as described in *Enabling access to the environment*.

Procedure

After you install the OpenShift Container Platform cluster, attach a floating IP address to the ingress port:

1. Show the port:

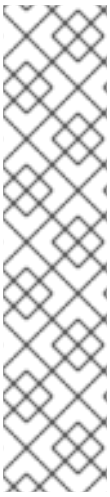
```
$ openstack port show <cluster name>-<clusterID>-ingress-port
```

2. Attach the port to the IP address:

```
$ openstack floating ip set --port <ingress port ID> <apps FIP>
```

3. Add a wildcard **A** record for ***apps.** to your DNS file:

```
*.apps.<cluster name>.<base domain>  IN  A  <apps FIP>
```



NOTE

If you do not control the DNS server but want to enable application access for non-production purposes, you can add these hostnames to **/etc/hosts**:

```
<apps FIP> console-openshift-console.apps.<cluster name>.<base domain>
<apps FIP> integrated-oauth-server-openshift-authentication.apps.<cluster name>.
<base domain>
<apps FIP> oauth-openshift.apps.<cluster name>.<base domain>
<apps FIP> prometheus-k8s-openshift-monitoring.apps.<cluster name>.<base
domain>
<apps FIP> grafana-openshift-monitoring.apps.<cluster name>.<base domain>
<apps FIP> <app name>.apps.<cluster name>.<base domain>
```

Next steps

- Customize your cluster.
- If necessary, you can opt out of remote health reporting .

1.3. UNINSTALLING A CLUSTER ON OPENSTACK

You can remove a cluster that you deployed to Red Hat OpenStack Platform (RHOSP).

1.3.1. Removing a cluster that uses installer-provisioned infrastructure

You can remove a cluster that uses installer-provisioned infrastructure from your cloud.

Prerequisites

- Have a copy of the installation program that you used to deploy the cluster.
- Have the files that the installation program generated when you created your cluster.

Procedure

1. From the computer that you used to install the cluster, run the following command:

```
$ ./openshift-install destroy cluster \
--dir=<installation_directory> --log-level=info 1 2
```

1. For **<installation_directory>**, specify the path to the directory that you stored the installation files in.
2. To view different details, specify **warn**, **debug**, or **error** instead of **info**.

NOTE

You must specify the directory that contains the cluster definition files for your cluster. The installation program requires the **metadata.json** file in this directory to delete the cluster.

2. Optional: Delete the **<installation_directory>** directory and the OpenShift Container Platform installation program.