OpenShift Container Platform 4.19

Installation overview

Overview content for installing OpenShift Container Platform

Last Updated: 2025-07-16

OpenShift Container Platform 4.19 Installation overview

Overview content for installing OpenShift Container Platform

Abstract

This document provides an overview on how to install OpenShift Container Platform.

Table of Contents

CHAPTER 1. OPENSHIFT CONTAINER PLATFORM INSTALLATION OVERVIEW

1.1. ABOUT OPENSHIFT CONTAINER PLATFORM INSTALLATION

The OpenShift Container Platform installation program offers four methods for deploying a cluster which are detailed in the following list:

- **Interactive**: You can deploy a cluster with the web-based Assisted Installer. This is an ideal approach for clusters with networks connected to the internet. The Assisted Installer is the easiest way to install OpenShift Container Platform, it provides smart defaults, and it performs pre-flight validations before installing the cluster. It also provides a RESTful API for automation and advanced configuration scenarios.
- **Local Agent-based**: You can deploy a cluster locally with the Agent-based Installer for disconnected environments or restricted networks. It provides many of the benefits of the Assisted Installer, but you must download and configure the Agent-based Installer first. Configuration is done with a command-line interface. This approach is ideal for disconnected environments.
- **Automated**: You can deploy a cluster on installer-provisioned infrastructure. The installation program uses each cluster host's baseboard management controller (BMC) for provisioning. You can deploy clusters in connected or disconnected environments.
- **Full control**: You can deploy a cluster on infrastructure that you prepare and maintain, which provides maximum customizability. You can deploy clusters in connected or disconnected environments.

Each method deploys a cluster with the following characteristics:

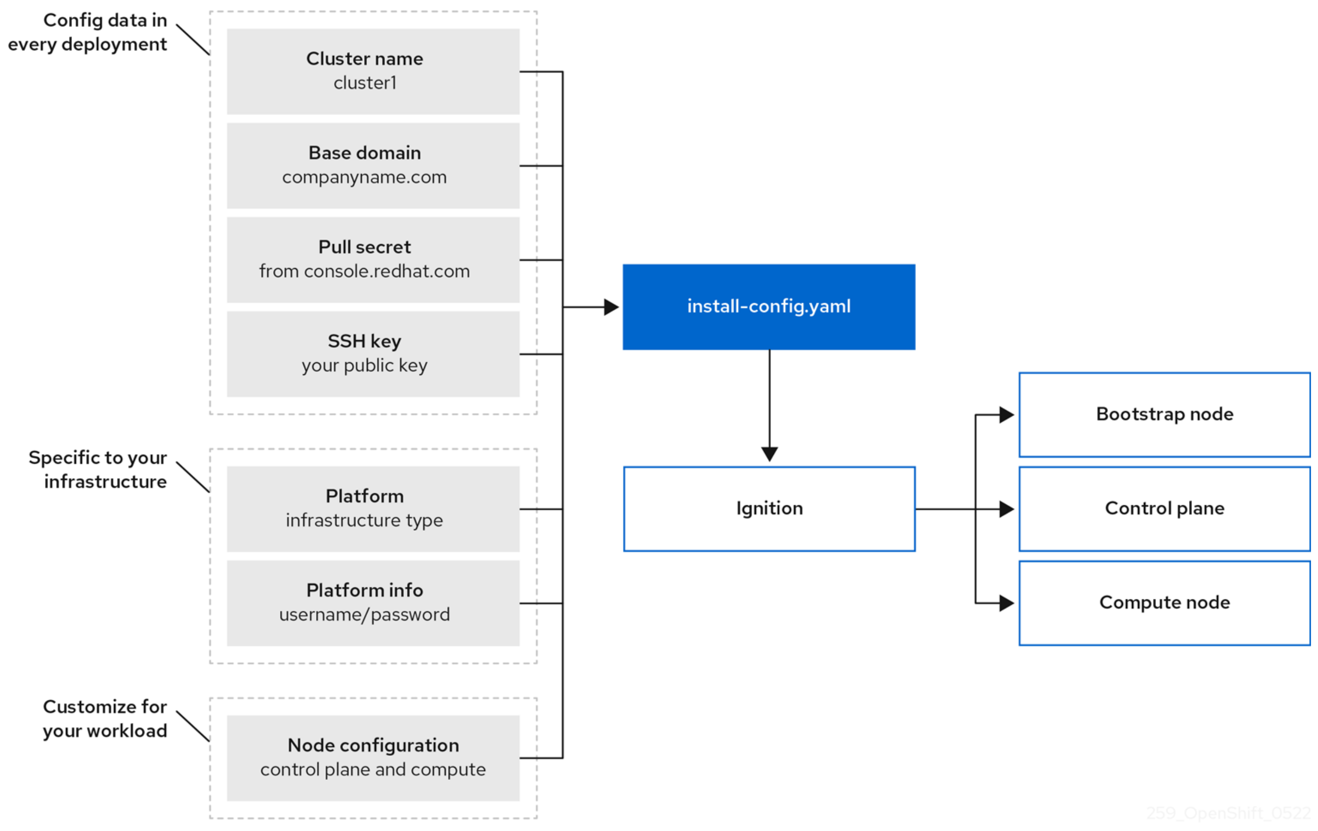
- Highly available infrastructure with no single points of failure, which is available by default.
- Administrators can control what updates are applied and when.

1.1.1. About the installation program

You can use the installation program to deploy each type of cluster. The installation program generates the main assets, such as Ignition config files for the bootstrap, control plane, and compute machines. You can start an OpenShift Container Platform cluster with these three machine configurations, provided you correctly configured the infrastructure.

The OpenShift Container Platform installation program uses a set of targets and dependencies to manage cluster installations. The installation program has a set of targets that it must achieve, and each target has a set of dependencies. Because each target is only concerned with its own dependencies, the installation program can act to achieve multiple targets in parallel with the ultimate target being a running cluster. The installation program recognizes and uses existing components instead of running commands to create them again because the program meets the dependencies.

Figure 1.1. OpenShift Container Platform installation targets and dependencies



1.1.2. About Red Hat Enterprise Linux CoreOS (RHCOS)

Post-installation, each cluster machine uses Red Hat Enterprise Linux CoreOS (RHCOS) as the operating system. RHCOS is the immutable container host version of Red Hat Enterprise Linux (RHEL) and features a RHEL kernel with SELinux enabled by default. RHCOS includes the **kubelet**, which is the Kubernetes node agent, and the CRI-O container runtime, which is optimized for Kubernetes.

Every control plane machine in an OpenShift Container Platform 4.19 cluster must use RHCOS, which includes a critical first-boot provisioning tool called Ignition. This tool enables the cluster to configure the machines. Operating system updates are delivered as a bootable container image, using **OSTree** as a backend, that is deployed across the cluster by the Machine Config Operator. Actual operating system changes are made in-place on each machine as an atomic operation by using **rpm-ostree**. Together, these technologies enable OpenShift Container Platform to manage the operating system like it manages any other application on the cluster, by in-place upgrades that keep the entire platform up to date. These in-place updates can reduce the burden on operations teams.

If you use RHCOS as the operating system for all cluster machines, the cluster manages all aspects of its components and machines, including the operating system. Because of this, only the installation program and the Machine Config Operator can change machines. The installation program uses Ignition config files to set the exact state of each machine, and the Machine Config Operator completes more changes to the machines, such as the application of new certificates or keys, after installation.

1.1.3. Glossary of common terms for OpenShift Container Platform installing

The glossary defines common terms that relate to the installation content. Read the following list of terms to better understand the installation process.

Assisted Installer

An installer hosted at console.redhat.com that provides a web-based user interface or a RESTful API for creating a cluster configuration. The Assisted Installer generates a discovery image. Cluster machines boot with the discovery image, which installs RHCOS and an agent. Together, the Assisted Installer and agent provide preinstallation validation and installation for the cluster.

Agent-based Installer

An installer similar to the Assisted Installer, but you must download the Agent-based Installer first. The Agent-based Installer is ideal for disconnected environments.

Bootstrap node

A temporary machine that runs a minimal Kubernetes configuration required to deploy the OpenShift Container Platform control plane.

Control plane

A container orchestration layer that exposes the API and interfaces to define, deploy, and manage the lifecycle of containers. Also known as control plane machines.

Compute node

Nodes that are responsible for executing workloads for cluster users. Also known as worker nodes.

Disconnected installation

In some situations, parts of a data center might not have access to the internet, even through proxy servers. You can still install the OpenShift Container Platform in these environments, but you must download the required software and images and make them available to the disconnected environment.

The OpenShift Container Platform installation program

A program that provisions the infrastructure and deploys a cluster.

Installer-provisioned infrastructure

The installation program deploys and configures the infrastructure that the cluster runs on.

Ignition config files

A file that the Ignition tool uses to configure Red Hat Enterprise Linux CoreOS (RHCOS) during operating system initialization. The installation program generates different Ignition configuration files to initialize bootstrap, control plane, and worker nodes.

Kubernetes manifests

Specifications of a Kubernetes API object in a JSON or YAML format. A configuration file can include deployments, config maps, secrets, daemonsets, and so on.

Kubelet

A primary node agent that runs on each node in the cluster to ensure that containers are running in a pod.

Load balancers

A load balancer serves as the single point of contact for clients. Load balancers for the API distribute incoming traffic across control plane nodes.

Machine Config Operator

An Operator that manages and applies configurations and updates of the base operating system and container runtime, including everything between the kernel and kubelet, for the nodes in the cluster.

Operators

The preferred method of packaging, deploying, and managing a Kubernetes application in an OpenShift Container Platform cluster. An operator takes human operational knowledge and encodes it into software that is easily packaged and shared with customers.

User-provisioned infrastructure

You can install OpenShift Container Platform on infrastructure that you provide. You can use the installation program to generate the assets required to provision the cluster infrastructure, create the cluster infrastructure, and then deploy the cluster to the infrastructure that you provided.

1.1.4. Installation process

Except for the Assisted Installer, when you install an OpenShift Container Platform cluster, you must download the installation program from the appropriate **Cluster Type** page on the OpenShift Cluster Manager Hybrid Cloud Console. This console manages:

- REST API for accounts.
- Registry tokens, which are the pull secrets that you use to obtain the required components.
- Cluster registration, which associates the cluster identity to your Red Hat account to facilitate the gathering of usage metrics.

In OpenShift Container Platform 4.19, the installation program is a Go binary file that performs a series of file transformations on a set of assets. The way you interact with the installation program differs depending on your installation type. Consider the following installation use cases:

- To deploy a cluster with the Assisted Installer, you must configure the cluster settings by using the Assisted Installer. There is no installation program to download and configure. After you finish setting the cluster configuration, you download a discovery ISO and then boot cluster machines with that image. You can install clusters with the Assisted Installer on Nutanix, vSphere, and bare metal with full integration, and other platforms without integration. If you install on bare metal, you must provide all of the cluster infrastructure and resources, including the networking, load balancing, storage, and individual cluster machines.
- To deploy clusters with the Agent-based Installer, you can download the Agent-based Installer first. You can then configure the cluster and generate a discovery image. You boot cluster machines with the discovery image, which installs an agent that communicates with the installation program and handles the provisioning for you instead of you interacting with the installation program or setting up a provisioner machine yourself. You must provide all of the cluster infrastructure and resources, including the networking, load balancing, storage, and individual cluster machines. This approach is ideal for disconnected environments.
- For clusters with installer-provisioned infrastructure, you delegate the infrastructure bootstrapping and provisioning to the installation program instead of doing it yourself. The installation program creates all of the networking, machines, and operating systems that are required to support the cluster, except if you install on bare metal. If you install on bare metal, you must provide all of the cluster infrastructure and resources, including the bootstrap machine, networking, load balancing, storage, and individual cluster machines.
- If you provision and manage the infrastructure for your cluster, you must provide all of the cluster infrastructure and resources, including the bootstrap machine, networking, load balancing, storage, and individual cluster machines.

For the installation program, the program uses three sets of files during installation: an installation configuration file that is named **install-config.yaml**, Kubernetes manifests, and Ignition config files for your machine types.

IMPORTANT

You can modify Kubernetes and the Ignition config files that control the underlying RHCOS operating system during installation. However, no validation is available to confirm the suitability of any modifications that you make to these objects. If you modify these objects, you might render your cluster non-functional. Because of this risk, modifying Kubernetes and Ignition config files is not supported unless you are following documented procedures or are instructed to do so by Red Hat support.

The installation configuration file is transformed into Kubernetes manifests, and then the manifests are wrapped into Ignition config files. The installation program uses these Ignition config files to create the cluster.

The installation configuration files are all pruned when you run the installation program, so be sure to back up all the configuration files that you want to use again.

IMPORTANT

You cannot modify the parameters that you set during installation, but you can modify many cluster attributes after installation.

The installation process with the Assisted Installer
Installation with the Assisted Installer involves creating a cluster configuration interactively by using the web-based user interface or the RESTful API. The Assisted Installer user interface prompts you for required values and provides reasonable default values for the remaining parameters, unless you change them in the user interface or with the API. The Assisted Installer generates a discovery image, which you download and use to boot the cluster machines. The image installs RHCOS and an agent, and the agent handles the provisioning for you. You can install OpenShift Container Platform with the Assisted Installer and full integration on Nutanix, vSphere, and bare metal. Additionally, you can install OpenShift Container Platform with the Assisted Installer on other platforms without integration.

OpenShift Container Platform manages all aspects of the cluster, including the operating system itself. Each machine boots with a configuration that references resources hosted in the cluster that it joins. This configuration allows the cluster to manage itself as updates are applied.

If possible, use the Assisted Installer feature to avoid having to download and configure the Agent-based Installer.

The installation process with Agent-based infrastructure
Agent-based installation is similar to using the Assisted Installer, except that you must initially download and install the Agent-based Installer. An Agent-based installation is useful when you want the convenience of the Assisted Installer, but you need to install a cluster in a disconnected environment.

If possible, use the Agent-based installation feature to avoid having to create a provisioner machine with a bootstrap VM, and then provision and maintain the cluster infrastructure.

The installation process with installer-provisioned infrastructure
The default installation type uses installer-provisioned infrastructure. By default, the installation program acts as an installation wizard, prompting you for values that it cannot determine on its own and providing reasonable default values for the remaining parameters. You can also customize the installation process to support advanced infrastructure scenarios. The installation program provisions the underlying infrastructure for the cluster.

You can install either a standard cluster or a customized cluster. With a standard cluster, you provide minimum details that are required to install the cluster. With a customized cluster, you can specify more details about the platform, such as the number of machines that the control plane uses, the type of virtual machine that the cluster deploys, or the CIDR range for the Kubernetes service network.

If possible, use this feature to avoid having to provision and maintain the cluster infrastructure. In all other environments, you use the installation program to generate the assets that you require to provision your cluster infrastructure.

With installer-provisioned infrastructure clusters, OpenShift Container Platform manages all aspects of the cluster, including the operating system itself. Each machine boots with a configuration that references resources hosted in the cluster that it joins. This configuration allows the cluster to manage itself as updates are applied.

The installation process with user-provisioned infrastructure
You can also install OpenShift Container Platform on infrastructure that you provide. You use the installation program to generate the assets that you require to provision the cluster infrastructure, create the cluster infrastructure, and then deploy the cluster to the infrastructure that you provided.

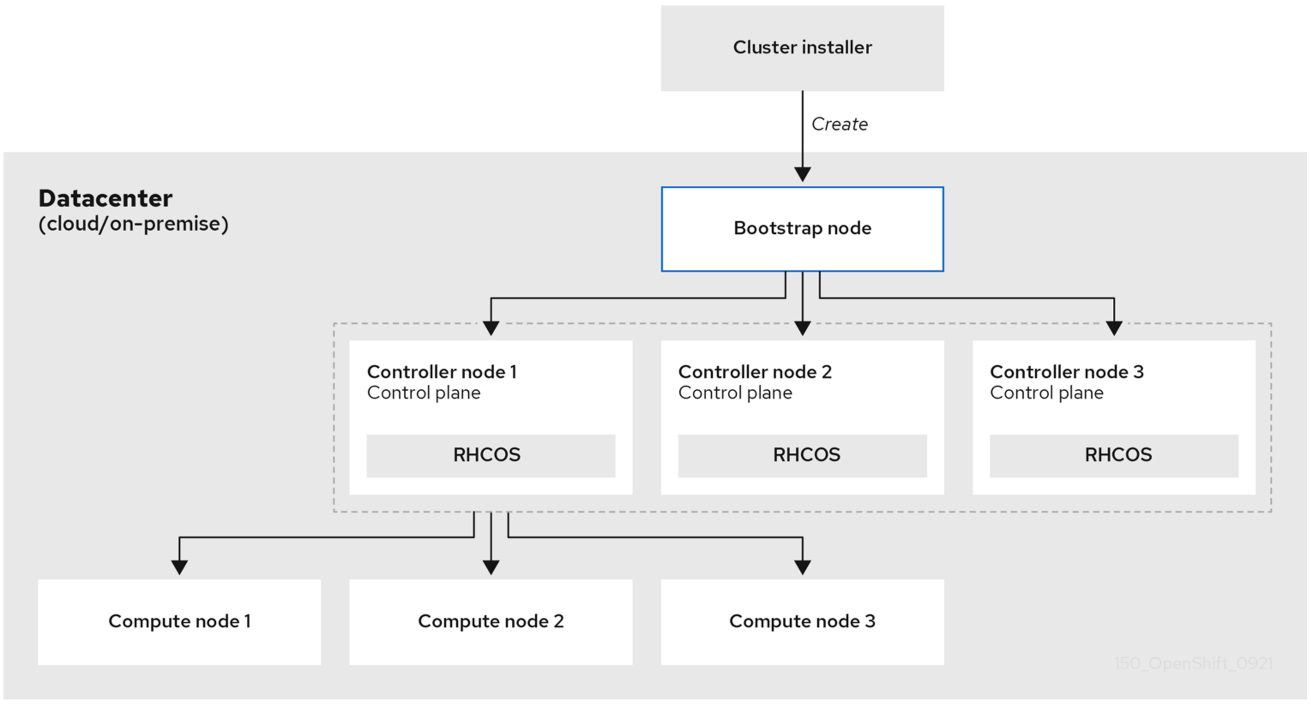
If you do not use infrastructure that the installation program provisioned, you must manage and maintain the cluster resources yourself. The following list details some of these self-managed resources:

- The underlying infrastructure for the control plane and compute machines that make up the cluster
- Load balancers
- Cluster networking, including the DNS records and required subnets
- Storage for the cluster infrastructure and applications

If your cluster uses user-provisioned infrastructure, you have the option of adding RHEL compute machines to your cluster.

Installation process details
When a cluster is provisioned, each machine in the cluster requires information about the cluster. OpenShift Container Platform uses a temporary bootstrap machine during initial configuration to provide the required information to the permanent control plane. The temporary bootstrap machine boots by using an Ignition config file that describes how to create the cluster. The bootstrap machine creates the control plane machines that make up the control plane. The control plane machines then create the compute machines, which are also known as worker machines. The following figure illustrates this process:

Figure 1.2. Creating the bootstrap, control plane, and compute machines



IMPORTANT

While planning to deploy your cluster, ensure that you are familiar with the recommended practices for performance and scalability, particularly the requirements for input/output (I/O) latency for etcd storage and the requirements for the recommended control plane node sizing. For more information, see “Recommended etcd practices” and “Control plane node sizing”.

After the cluster machines initialize, the bootstrap machine is destroyed. All clusters use the bootstrap process to initialize the cluster, but if you provision the infrastructure for your cluster, you must complete many of the steps manually.

IMPORTANT

- The Ignition config files that the installation program generates contain certificates that expire after 24 hours, which are then renewed at that time. If the cluster is shut down before renewing the certificates and the cluster is later restarted after the 24 hours have elapsed, the cluster automatically recovers the expired certificates. The exception is that you must manually approve the pending **node-bootstrapper** certificate signing requests (CSRs) to recover kubelet certificates. See the documentation for *Recovering from expired control plane certificates* for more information.

- Consider using Ignition config files within 12 hours after they are generated, because the 24-hour certificate rotates from 16 to 22 hours after the cluster is installed. By using the Ignition config files within 12 hours, you can avoid installation failure if the certificate update runs during installation.

Bootstrapping a cluster involves the following steps:

1. The bootstrap machine boots and starts hosting the remote resources required for the control plane machines to boot. If you provision the infrastructure, this step requires manual intervention.

2. The bootstrap machine starts a single-node etcd cluster and a temporary Kubernetes control plane.

3. The control plane machines fetch the remote resources from the bootstrap machine and finish booting. If you provision the infrastructure, this step requires manual intervention.

4. The temporary control plane schedules the production control plane to the production control plane machines.

5. The Cluster Version Operator (CVO) comes online and installs the etcd Operator. The etcd Operator scales up etcd on all control plane nodes.

6. The temporary control plane shuts down and passes control to the production control plane.

7. The bootstrap machine injects OpenShift Container Platform components into the production control plane.

8. The installation program shuts down the bootstrap machine. If you provision the infrastructure, this step requires manual intervention.

9. The control plane sets up the compute nodes.

10. The control plane installs additional services in the form of a set of Operators.

The result of this bootstrapping process is a running OpenShift Container Platform cluster. The cluster then downloads and configures remaining components needed for the day-to-day operations, including the creation of compute machines in supported environments.

Additional resources

- Recommended etcd practices

- Control plane node sizing

- Red Hat OpenShift Network Calculator

1.1.5. Verifying node state after installation

The OpenShift Container Platform installation completes when the following installation health checks are successful:

- The provisioner can access the OpenShift Container Platform web console.

- All control plane nodes are ready.

- All cluster Operators are available.

NOTE

After the installation completes, the specific cluster Operators responsible for the worker nodes continuously attempt to provision all worker nodes. Some time is required before all worker nodes report as **READY**. For installations on bare metal, wait a minimum of 60 minutes before troubleshooting a worker node. For installations on all other platforms, wait a minimum of 40 minutes before troubleshooting a worker node. A **DEGRADED** state for the cluster Operators responsible for the worker nodes depends on the Operators' own resources and not on the state of the nodes.

After your installation completes, you can continue to monitor the condition of the nodes in your cluster.

Prerequisites

- The installation program resolves successfully in the terminal.

Procedure

1. Show the status of all worker nodes:

```
$ oc get nodes
```

Example output

```
NAME                         STATUS   ROLES    AGE   VERSION
example-compute1.example.com   Ready    worker   13m   v1.21.6+bb8d50a
example-compute2.example.com   Ready    worker   13m   v1.21.6+bb8d50a
example-compute4.example.com   Ready    worker   14m   v1.21.6+bb8d50a
example-control1.example.com   Ready    master   52m   v1.21.6+bb8d50a
example-control2.example.com   Ready    master   55m   v1.21.6+bb8d50a
example-control3.example.com   Ready    master   55m   v1.21.6+bb8d50a
```

2. Show the phase of all worker machine nodes:

```
$ oc get machines -A
```

Example output

```
NAMESPACE             NAME                        PHASE       TYPE   REGION   ZONE   AGE
openshift-machine-api   example-zbbt6-master-0        Running                              95m
openshift-machine-api   example-zbbt6-master-1        Running                              95m
openshift-machine-api   example-zbbt6-master-2        Running                              95m
openshift-machine-api   example-zbbt6-worker-0-25bhp   Running                              49m
openshift-machine-api   example-zbbt6-worker-0-8b4c2   Running                              49m
openshift-machine-api   example-zbbt6-worker-0-jkbqt   Running                              49m
openshift-machine-api   example-zbbt6-worker-0-qrl5b   Running                              49m
```

Additional resources

- Getting the BareMetalHost resource
- Following the progress of the installation

- Validating an installation
- Agent-based Installer
- Assisted Installer for OpenShift Container Platform

Installation scope

The scope of the OpenShift Container Platform installation program is intentionally narrow. It is designed for simplicity and ensured success. You can complete many more configuration tasks after installation completes.

Additional resources

- See Available cluster customizations for details about OpenShift Container Platform configuration resources.

1.1.6. OpenShift Local overview

OpenShift Local supports rapid application development to get started building OpenShift Container Platform clusters. OpenShift Local is designed to run on a local computer to simplify setup and testing, and to emulate the cloud development environment locally with all of the tools needed to develop container-based applications.

Regardless of the programming language you use, OpenShift Local hosts your application and brings a minimal, preconfigured Red Hat OpenShift Container Platform cluster to your local PC without the need for a server-based infrastructure.

On a hosted environment, OpenShift Local can create microservices, convert them into images, and run them in Kubernetes-hosted containers directly on your laptop or desktop running Linux, macOS, or Windows 10 or later.

For more information about OpenShift Local, see Red Hat OpenShift Local Overview .

1.2. SUPPORTED PLATFORMS FOR OPENSHIFT CONTAINER PLATFORM CLUSTERS

The following table describes which platforms are supported by the different methods available for installing OpenShift Container Platform clusters:

Table 1.1. Supported platforms

Platform	Installer-provisioned infrastructure [1]	User-provisioned infrastructure [2]	Agent-based Installer	Assisted Installer
Amazon Web Services (AWS)	X	X		
Bare metal	X	X	X	X
External			X	X

Platform	Installer-provisioned infrastructure [1]	User-provisioned infrastructure [2]	Agent-based Installer	Assisted Installer
Google Cloud Platform (GCP)	X	X		
IBM Cloud® Classic	X			
IBM Cloud® Virtual Private Cloud (VPC)	X			
IBM Power®		X	X	X
IBM Z® or IBM® LinuxONE		X	X	X
Microsoft Azure	X	X		
Microsoft Azure Stack Hub	X	X		
None			X	X
Nutanix	X			X
Oracle Cloud Infrastructure (OCI)			X	X
Red Hat OpenStack Platform (RHOSP) [3]	X	X		
VMware vSphere	X	X	X	X

1. For installer-provisioned infrastructure: All machines, including the computer that you run the installation process on, must have direct internet access to pull images for platform containers and provide telemetry data to Red Hat.

   IMPORTANT

   After installation, the following changes are not supported:

   - Mixing cloud provider platforms.
   - Mixing cloud provider components. For example, using a persistent storage framework from a another platform on the platform where you installed the cluster.

2. For user-provisioned infrastructure: Depending on the supported cases for the platform, you can perform installations on user-provisioned infrastructure so that you can run machines with full internet access, place your cluster behind a proxy, or perform a disconnected installation.
   In a disconnected installation, you can download the images that are required to install a cluster, place them in a mirror registry, and use that data to install your cluster. While you require

internet access to pull images for platform containers, with a disconnected installation on vSphere or bare-metal infrastructure, your cluster machines do not require direct internet access.

3. For Red Hat OpenStack Platform (RHOSP): The latest OpenShift Container Platform release supports both the latest RHOSP long-life release and intermediate release. For complete RHOSP release compatibility, see the OpenShift Container Platform on RHOSP support matrix .

The OpenShift Container Platform 4.x Tested Integrations page contains details about integration testing for different platforms.

Additional resources

- See Supported installation methods for different platforms for more information about the types of installations that are available for each supported platform.
- See Selecting a cluster installation method and preparing it for users for information about choosing an installation method and preparing the required resources.
- Red Hat OpenShift Network Calculator can help you design your cluster network during both the deployment and expansion phases. It addresses common questions related to the cluster network and provides output in a convenient JSON format.

CHAPTER 2. SELECTING A CLUSTER INSTALLATION METHOD AND PREPARING IT FOR USERS

Before you install OpenShift Container Platform, decide what kind of installation process to follow and verify that you have all of the required resources to prepare the cluster for users.

2.1. SELECTING A CLUSTER INSTALLATION TYPE

Before you install an OpenShift Container Platform cluster, you need to select the best installation instructions to follow. Think about your answers to the following questions to select the best option.

2.1.1. Do you want to install and manage an OpenShift Container Platform cluster yourself?

If you want to install and manage OpenShift Container Platform yourself, you can install it on the following platforms:

- Amazon Web Services (AWS) on 64-bit x86 instances
- Amazon Web Services (AWS) on 64-bit ARM instances
- Microsoft Azure on 64-bit x86 instances
- Microsoft Azure on 64-bit ARM instances
- Microsoft Azure Stack Hub
- Google Cloud Platform (GCP) on 64-bit x86 instances
- Google Cloud Platform (GCP) on 64-bit ARM instances
- Red Hat OpenStack Platform (RHOSP)
- IBM Cloud®
- IBM Z® or IBM® LinuxONE with z/VM
- IBM Z® or IBM® LinuxONE with Red Hat Enterprise Linux (RHEL) KVM
- IBM Z® or IBM® LinuxONE in an LPAR
- IBM Power®
- IBM Power® Virtual Server
- Nutanix
- VMware vSphere
- Bare metal or other platform agnostic infrastructure

You can deploy an OpenShift Container Platform 4 cluster to both on-premise hardware and to cloud hosting services, but all of the machines in a cluster must be in the same data center or cloud hosting service.

If you want to use OpenShift Container Platform but you do not want to manage the cluster yourself, you can choose from several managed service options. If you want a cluster that is fully managed by Red Hat, you can use OpenShift Dedicated. You can also use OpenShift as a managed service on Azure, AWS, IBM Cloud®, or Google Cloud Platform. For more information about managed services, see the OpenShift Products page. If you install an OpenShift Container Platform cluster with a cloud virtual machine as a virtual bare metal, the corresponding cloud-based storage is not supported.

2.1.2. Have you used OpenShift Container Platform 3 and want to use OpenShift Container Platform 4?

If you used OpenShift Container Platform 3 and want to try OpenShift Container Platform 4, you need to understand how different OpenShift Container Platform 4 is. OpenShift Container Platform 4 weaves the Operators that package, deploy, and manage Kubernetes applications and the operating system that the platform runs on, Red Hat Enterprise Linux CoreOS (RHCOS), together seamlessly. Instead of deploying machines and configuring their operating systems so that you can install OpenShift Container Platform on them, the RHCOS operating system is an integral part of the OpenShift Container Platform cluster. Deploying the operating system for the cluster machines is part of the installation process for OpenShift Container Platform. See Differences between OpenShift Container Platform 3 and 4 .

Because you need to provision machines as part of the OpenShift Container Platform cluster installation process, you cannot upgrade an OpenShift Container Platform 3 cluster to OpenShift Container Platform 4. Instead, you must create a new OpenShift Container Platform 4 cluster and migrate your OpenShift Container Platform 3 workloads to them. For more information about migrating, see Migrating from OpenShift Container Platform 3 to 4 overview . Because you must migrate to OpenShift Container Platform 4, you can use any type of production cluster installation process to create your new cluster.

2.1.3. Do you want to use existing components in your cluster?

Because the operating system is integral to OpenShift Container Platform, it is easier to let the installation program for OpenShift Container Platform stand up all of the infrastructure. These are called *installer provisioned infrastructure* installations. In this type of installation, you can provide some existing infrastructure to the cluster, but the installation program deploys all of the machines that your cluster initially needs.

You can deploy an installer-provisioned infrastructure cluster without specifying any customizations to the cluster or its underlying machines to AWS, Azure, Azure Stack Hub, GCP, Nutanix.

If you need to perform basic configuration for your installer-provisioned infrastructure cluster, such as the instance type for the cluster machines, you can customize an installation for AWS, Azure, GCP, Nutanix.

For installer-provisioned infrastructure installations, you can use an existing VPC in AWS, vNet in Azure, or VPC in GCP. You can also reuse part of your networking infrastructure so that your cluster in AWS, Azure, GCP can coexist with existing IP address allocations in your environment and integrate with existing MTU and VXLAN configurations. If you have existing accounts and credentials on these clouds, you can re-use them, but you might need to modify the accounts to have the required permissions to install OpenShift Container Platform clusters on them.

You can use the installer-provisioned infrastructure method to create appropriate machine instances on your hardware for vSphere, and bare metal. Additionally, for vSphere, you can also customize additional network parameters during installation.

For some installer-provisioned infrastructure installations, for example on the VMware vSphere and bare metal platforms, the external traffic that reaches the ingress virtual IP (VIP) is not balanced between the default **IngressController** replicas. For vSphere and bare metal installer-provisioned infrastructure installations where exceeding the baseline **IngressController** router performance is expected, you must configure an external load balancer. Configuring an external load balancer achieves the performance of multiple **IngressController** replicas. For more information about the baseline **IngressController** performance, see Baseline Ingress Controller (router) performance . For more information about configuring an external load balancer, see Configuring a user-managed load balancer .

If you want to reuse extensive cloud infrastructure, you can complete a *user-provisioned infrastructure* installation. With these installations, you manually deploy the machines that your cluster requires during the installation process. If you perform a user-provisioned infrastructure installation on AWS, Azure, Azure Stack Hub , you can use the provided templates to help you stand up all of the required components. You can also reuse a shared VPC on GCP. Otherwise, you can use the provider-agnostic installation method to deploy a cluster into other clouds.

You can also complete a user-provisioned infrastructure installation on your existing hardware. If you use RHOSP, IBM Z® or IBM® LinuxONE, IBM Z® and IBM® LinuxONE with RHEL KVM , IBM Z® and IBM® LinuxONE in an LPAR, IBM Power, or vSphere, use the specific installation instructions to deploy your cluster. If you use other supported hardware, follow the bare metal installation procedure. For some of these platforms, such as vSphere, and bare metal, you can also customize additional network parameters during installation.

2.1.4. Do you need extra security for your cluster?

If you use a user-provisioned installation method, you can configure a proxy for your cluster. The instructions are included in each installation procedure.

If you want to prevent your cluster on a public cloud from exposing endpoints externally, you can deploy a private cluster with installer-provisioned infrastructure on AWS, Azure, or GCP.

If you need to install your cluster that has limited access to the internet, such as a disconnected or restricted network cluster, you can mirror the installation packages and install the cluster from them. Follow detailed instructions for user-provisioned infrastructure installations into restricted networks for AWS, GCP, IBM Z® or IBM® LinuxONE, IBM Z® or IBM® LinuxONE with RHEL KVM , IBM Z® or IBM® LinuxONE in an LPAR, IBM Power® , vSphere, or bare metal. You can also install a cluster into a restricted network using installer-provisioned infrastructure by following detailed instructions for AWS, GCP, IBM Cloud®, Nutanix, RHOSP, and vSphere.

If you need to deploy your cluster to an AWS GovCloud region, AWS China region, or Azure government region, you can configure those custom regions during an installer-provisioned infrastructure installation.

You can also configure the cluster machines to use the RHEL cryptographic libraries that have been submitted to NIST for FIPS 140-2/140-3 Validation during installation.

IMPORTANT

When running Red Hat Enterprise Linux (RHEL) or Red Hat Enterprise Linux CoreOS (RHCOS) booted in FIPS mode, OpenShift Container Platform core components use the RHEL cryptographic libraries that have been submitted to NIST for FIPS 140-2/140-3 Validation on only the x86_64, ppc64le, and s390x architectures.

2.2. PREPARING YOUR CLUSTER FOR USERS AFTER INSTALLATION

Some configuration is not required to install the cluster but recommended before your users access the cluster. You can customize the cluster itself by customizing the Operators that make up your cluster and integrate you cluster with other required systems, such as an identity provider.

For a production cluster, you must configure the following integrations:

- Persistent storage
- An identity provider
- Monitoring core OpenShift Container Platform components

2.3. PREPARING YOUR CLUSTER FOR WORKLOADS

Depending on your workload needs, you might need to take extra steps before you begin deploying applications. For example, after you prepare infrastructure to support your application build strategy, you might need to make provisions for low-latency workloads or to protect sensitive workloads. You can also configure monitoring for application workloads.

2.4. SUPPORTED INSTALLATION METHODS FOR DIFFERENT PLATFORMS

You can perform different types of installations on different platforms.

NOTE

Not all installation options are supported for all platforms, as shown in the following tables. A checkmark indicates that the option is supported and links to the relevant section.

Table 2.1. Installer-provisioned infrastructure options

	AWS (64-bit x86)	AWS (64-bit ARM)	Azure (64-bit x86)	Azure (64-bit ARM)	Azure Stack Hub	GCP (64-bit x86)	GCP (64-bit ARM)	Nutanix	RHOSP	Bare metal (64-bit x86)	Bare metal (64-bit ARM)	vSphere	IBM Cloud®	IBM Z®	IBM Power®	IBM Power® Virtual Server
Default	✓	✓	✓	✓	✓	✓	✓	✓		✓	✓	✓	✓			
Custom	✓	✓	✓	✓	✓	✓	✓	✓	✓			✓	✓			✓

	AWS (64-bit x86)	AWS (64-bit ARM)	Azure (64-bit x86)	Azure (64-bit ARM)	Azure Stack Hub	GCP (64-bit x86)	GCP (64-bit ARM)	Nutanix	RHOSP	Bare metal (64-bit x86)	Bare metal (64-bit ARM)	vSphere	IBM Cloud®	IBM Z®	IBM Power®	IBM Power® Virtual Server
Network customization	✓	✓	✓	✓	✓	✓	✓			✓	✓	✓	✓			
Restricted network	✓	✓	✓	✓		✓	✓	✓	✓	✓	✓	✓	✓			✓
Private clusters	✓	✓	✓	✓		✓	✓						✓			✓

	AWS (64-bit x86)	AWS (64-bit ARM)	Azure (64-bit x86)	Azure (64-bit ARM)	Azure Stack Hub	GCP (64-bit x86)	GCP (64-bit ARM)	Nutanix	RHOSP	Bare metal (64-bit x86)	Bare metal (64-bit ARM)	vSphere	IBM Cloud®	IBM Z®	IBM Power®	IBM Power® Virtual Server
Existing virtual private networks	✓	✓	✓	✓		✓	✓						✓			✓
Government regions	✓		✓													
Secret regions	✓															
China regions	✓															

Table 2.2. User-provisioned infrastructure options

	AWS (64-bit x86)	AWS (64-bit ARM)	Azure (64-bit x86)	Azure (64-bit ARM)	Azure Stack Hub	GCP (64-bit x86)	GCP (64-bit ARM)	Nutanix	RHOSP	Bare metal (64-bit x86)	Bare metal (64-bit ARM)	vSphere	IBM Cloud®	IBM Z®	IBM Z® with RHEL KVM	IBM Power®	Platform agnostic
Custom	✓	✓	✓	✓	✓	✓	✓		✓	✓	✓	✓		✓	✓	✓	✓
Network customization										✓	✓	✓					
Restricted network	✓	✓				✓	✓			✓	✓	✓		✓	✓	✓	

	AWS (64-bit x86)	AWS (64-bit ARM)	Azure (64-bit x86)	Azure (64-bit ARM)	Azure Stack Hub	GCP (64-bit x86)	GCP (64-bit ARM)	Nutanix	RHOSP	Bare metal (64-bit x86)	Bare metal (64-bit ARM)	vSphere	IBM Cloud®	IBM Z®	IBM Z® with RHEL KVM	IBM Power®	Platform agnostic
Shared VPC hosted outside of cluster project						✓	✓										

CHAPTER 3. CLUSTER CAPABILITIES

Cluster administrators can use cluster capabilities to enable or disable optional components prior to installation. Cluster administrators can enable cluster capabilities at anytime after installation.

NOTE

Cluster administrators cannot disable a cluster capability after it is enabled.

3.1. ENABLING CLUSTER CAPABILITIES

If you are using an installation method that includes customizing your cluster by creating an **install-config.yaml** file, you can select which cluster capabilities you want to make available on the cluster.

NOTE

If you customize your cluster by enabling or disabling specific cluster capabilities, you must manually maintain your **install-config.yaml** file. New OpenShift Container Platform updates might declare new capability handles for existing components, or introduce new components altogether. Users who customize their **install-config.yaml** file should consider periodically updating their **install-config.yaml** file as OpenShift Container Platform is updated.

You can use the following configuration parameters to select cluster capabilities:

```
capabilities:
  baselineCapabilitySet: v4.11 (1)
  additionalEnabledCapabilities: (2)
  - CSISnapshot
  - Console
  - Storage
```

(1) Defines a baseline set of capabilities to install. Valid values are **None**, **vCurrent** and **v4.x**. If you select **None**, all optional capabilities are disabled. The default value is **vCurrent**, which enables all optional capabilities.

NOTE

v4.x refers to any value up to and including the current cluster version. For example, valid values for a OpenShift Container Platform 4.12 cluster are **v4.11** and **v4.12**.

(2) Defines a list of capabilities to explicitly enable. These capabilities are enabled in addition to the capabilities specified in **baselineCapabilitySet**.

NOTE

In this example, the default capability is set to **v4.11**. The **additionalEnabledCapabilities** field enables additional capabilities over the default **v4.11** capability set.

The following table describes the **baselineCapabilitySet** values.

Table 3.1. Cluster capabilities **baselineCapabilitySet** values description

Value	Description
vCurrent	Specify this option when you want to automatically add new, default capabilities that are introduced in new releases.
v4.11	Specify this option when you want to enable the default capabilities for OpenShift Container Platform 4.11. By specifying **v4.11**, capabilities that are introduced in newer versions of OpenShift Container Platform are not enabled. The default capabilities in OpenShift Container Platform 4.11 are **baremetal**, **MachineAPI**, **marketplace**, and **openshift-samples**.
v4.12	Specify this option when you want to enable the default capabilities for OpenShift Container Platform 4.12. By specifying **v4.12**, capabilities that are introduced in newer versions of OpenShift Container Platform are not enabled. The default capabilities in OpenShift Container Platform 4.12 are **baremetal**, **MachineAPI**, **marketplace**, **openshift-samples**, **Console**, **Insights**, **Storage**, and **CSISnapshot**.
v4.13	Specify this option when you want to enable the default capabilities for OpenShift Container Platform 4.13. By specifying **v4.13**, capabilities that are introduced in newer versions of OpenShift Container Platform are not enabled. The default capabilities in OpenShift Container Platform 4.13 are **baremetal**, **MachineAPI**, **marketplace**, **openshift-samples**, **Console**, **Insights**, **Storage**, **CSISnapshot**, and **NodeTuning**.
v4.14	Specify this option when you want to enable the default capabilities for OpenShift Container Platform 4.14. By specifying **v4.14**, capabilities that are introduced in newer versions of OpenShift Container Platform are not enabled. The default capabilities in OpenShift Container Platform 4.14 are **baremetal**, **MachineAPI**, **marketplace**, **openshift-samples**, **Console**, **Insights**, **Storage**, **CSISnapshot**, **NodeTuning**, **ImageRegistry**, **Build**, and **DeploymentConfig**.
v4.15	Specify this option when you want to enable the default capabilities for OpenShift Container Platform 4.15. By specifying **v4.15**, capabilities that are introduced in newer versions of OpenShift Container Platform are not enabled. The default capabilities in OpenShift Container Platform 4.15 are **baremetal**, **MachineAPI**, **marketplace**, **OperatorLifecycleManager**, **openshift-samples**, **Console**, **Insights**, **Storage**, **CSISnapshot**, **NodeTuning**, **ImageRegistry**, **Build**, **CloudCredential**, and **DeploymentConfig**.

Value	Description
v4.16	Specify this option when you want to enable the default capabilities for OpenShift Container Platform 4.16. By specifying **v4.16**, capabilities that are introduced in newer versions of OpenShift Container Platform are not enabled. The default capabilities in OpenShift Container Platform 4.16 are **baremetal**, **MachineAPI**, **marketplace**, **OperatorLifecycleManager**, **openshift-samples**, **Console**, **Insights**, **Storage**, **CSISnapshot**, **NodeTuning**, **ImageRegistry**, **Build**, **CloudCredential**, **DeploymentConfig**, and **CloudControllerManager**.
v4.17	Specify this option when you want to enable the default capabilities for OpenShift Container Platform 4.17. By specifying **v4.17**, capabilities that are introduced in newer versions of OpenShift Container Platform are not enabled. The default capabilities in OpenShift Container Platform 4.17 are **baremetal**, **MachineAPI**, **marketplace**, **OperatorLifecycleManager**, **openshift-samples**, **Console**, **Insights**, **Storage**, **CSISnapshot**, **NodeTuning**, **ImageRegistry**, **Build**, **CloudCredential**, **DeploymentConfig**, and **CloudControllerManager**.
v4.18	Specify this option when you want to enable the default capabilities for OpenShift Container Platform 4.18. By specifying **v4.18**, capabilities that are introduced in newer versions of OpenShift Container Platform are not enabled. The default capabilities in OpenShift Container Platform 4.18 are **baremetal**, **MachineAPI**, **marketplace**, **OperatorLifecycleManager**, **OperatorLifecycleManagerV1**, **openshift-samples**, **Console**, **Insights**, **Storage**, **CSISnapshot**, **NodeTuning**, **ImageRegistry**, **Build**, **CloudCredential**, **DeploymentConfig**, and **CloudControllerManager**.
None	Specify when the other sets are too large, and you do not need any capabilities or want to fine-tune via **additionalEnabledCapabilities**.

Additional resources

- Installing a cluster on AWS with customizations
- Installing a cluster on GCP with customizations

3.2. OPTIONAL CLUSTER CAPABILITIES IN OPENSHIFT CONTAINER PLATFORM 4.19

Currently, cluster Operators provide the features for these optional capabilities. The following summarizes the features provided by each capability and what functionality you lose if it is disabled.

Additional resources

- Cluster Operators reference

3.2.1. Bare-metal capability

Purpose
The Cluster Baremetal Operator provides the features for the **baremetal** capability.

The Cluster Baremetal Operator (CBO) deploys all the components necessary to take a bare-metal server to a fully functioning worker node ready to run OpenShift Container Platform compute nodes. The CBO ensures that the metal3 deployment, which consists of the Bare Metal Operator (BMO) and Ironic containers, runs on one of the control plane nodes within the OpenShift Container Platform cluster. The CBO also listens for OpenShift Container Platform updates to resources that it watches and takes appropriate action.

The bare-metal capability is required for deployments using installer-provisioned infrastructure. Disabling the bare-metal capability can result in unexpected problems with these deployments.

IMPORTANT

If the bare-metal capability is disabled, the cluster cannot provision or manage bare-metal nodes. Only disable the capability if there are no **BareMetalHost** resources in your deployment. The **baremetal** capability depends on the **MachineAPI** capability. If you enable the **baremetal** capability, you must also enable **MachineAPI**.

NOTE

It is recommended that cluster administrators only disable the bare-metal capability during installations with user-provisioned infrastructure that do not have any **BareMetalHost** resources in the cluster.

Additional resources

- Deploying installer-provisioned clusters on bare metal
- Preparing for bare metal cluster installation
- Configuration using the Bare Metal Operator

3.2.2. Build capability

Purpose
The **Build** capability enables the **Build** API. The **Build** API manages the lifecycle of **Build** and **BuildConfig** objects.

IMPORTANT

If you disable the **Build** capability, the following resources will not be available in the cluster:

- **Build** and **BuildConfig** resources
- The **builder** service account

Disable the **Build** capability only if you do not require **Build** and **BuildConfig** resources or the **builder** service account in the cluster.

3.2.3. Cloud controller manager capability

Purpose
The Cloud Controller Manager Operator provides features for the **CloudControllerManager** capability.

NOTE

Currently, disabling the **CloudControllerManager** capability is not supported on all platforms.

You can determine if your cluster supports disabling the **CloudControllerManager** capability by checking values in the installation configuration (**install-config.yaml**) file for your cluster.

In the **install-config.yaml** file, locate the **platform** parameter.

- If the value of the **platform** parameter is **Baremetal** or **None**, you can disable the **CloudControllerManager** capability on your cluster.
- If the value of the **platform** parameter is **External**, locate the **platform.external.cloudControllerManager** parameter. If the value of the **platform.external.cloudControllerManager** parameter is **None**, you can disable the **CloudControllerManager** capability on your cluster.

IMPORTANT

If these parameters contain any other values than those listed, you cannot disable the **CloudControllerManager** capability on your cluster.

NOTE

The status of this Operator is General Availability for Amazon Web Services (AWS), Google Cloud Platform (GCP), IBM Cloud®, global Microsoft Azure, Microsoft Azure Stack Hub, Nutanix, Red Hat OpenStack Platform (RHOSP), and VMware vSphere.

The Operator is available as a Technology Preview for IBM Power® Virtual Server.

The Cloud Controller Manager Operator manages and updates the cloud controller managers deployed on top of OpenShift Container Platform. The Operator is based on the Kubebuilder framework and **controller-runtime** libraries. It is installed via the Cluster Version Operator (CVO).

It contains the following components:

- Operator

- Cloud configuration observer

By default, the Operator exposes Prometheus metrics through the **metrics** service.

3.2.4. Cloud credential capability

Purpose
The Cloud Credential Operator provides features for the **CloudCredential** capability.

NOTE

Currently, disabling the **CloudCredential** capability is only supported for bare-metal clusters.

The Cloud Credential Operator (CCO) manages cloud provider credentials as Kubernetes custom resource definitions (CRDs). The CCO syncs on **CredentialsRequest** custom resources (CRs) to allow OpenShift Container Platform components to request cloud provider credentials with the specific permissions that are required for the cluster to run.

By setting different values for the **credentialsMode** parameter in the **install-config.yaml** file, the CCO can be configured to operate in several different modes. If no mode is specified, or the **credentialsMode** parameter is set to an empty string (**""**), the CCO operates in its default mode.

Additional resources

- About the Cloud Credential Operator

3.2.5. Cluster Image Registry capability

Purpose
The Cluster Image Registry Operator provides features for the **ImageRegistry** capability.

The Cluster Image Registry Operator manages a singleton instance of the OpenShift image registry. It manages all configuration of the registry, including creating storage.

On initial start up, the Operator creates a default **image-registry** resource instance based on the configuration detected in the cluster. This indicates what cloud storage type to use based on the cloud provider.

If insufficient information is available to define a complete **image-registry** resource, then an incomplete resource is defined and the Operator updates the resource status with information about what is missing.

The Cluster Image Registry Operator runs in the **openshift-image-registry** namespace and it also manages the registry instance in that location. All configuration and workload resources for the registry reside in that namespace.

In order to integrate the image registry into the cluster's user authentication and authorization system, an image pull secret is generated for each service account in the cluster.

IMPORTANT

If you disable the **ImageRegistry** capability or if you disable the integrated OpenShift image registry in the Cluster Image Registry Operator's configuration, the image pull secret is not generated for each service account.

If you disable the **ImageRegistry** capability, you can reduce the overall resource footprint of OpenShift Container Platform in Telco environments. Depending on your deployment, you can disable this component if you do not need it.

Project
cluster-image-registry-operator

Additional resources

- Image Registry Operator in OpenShift Container Platform
- Automatically generated secrets

3.2.6. Cluster storage capability

Purpose
The Cluster Storage Operator provides the features for the **Storage** capability.

The Cluster Storage Operator sets OpenShift Container Platform cluster-wide storage defaults. It ensures a default **storageclass** exists for OpenShift Container Platform clusters. It also installs Container Storage Interface (CSI) drivers which enable your cluster to use various storage backends.

IMPORTANT

If the cluster storage capability is disabled, the cluster will not have a default **storageclass** or any CSI drivers. Users with administrator privileges can create a default **storageclass** and manually install CSI drivers if the cluster storage capability is disabled.

Notes

- The storage class that the Operator creates can be made non-default by editing its annotation, but this storage class cannot be deleted as long as the Operator runs.

3.2.7. Console capability

Purpose
The Console Operator provides the features for the **Console** capability.

The Console Operator installs and maintains the OpenShift Container Platform web console on a cluster. The Console Operator is installed by default and automatically maintains a console.

Additional resources

- Web console overview

3.2.8. CSI snapshot controller capability

Purpose
The Cluster CSI Snapshot Controller Operator provides the features for the **CSISnapshot** capability.

The Cluster CSI Snapshot Controller Operator installs and maintains the CSI Snapshot Controller. The CSI Snapshot Controller is responsible for watching the **VolumeSnapshot** CRD objects and manages the creation and deletion lifecycle of volume snapshots.

Additional resources

Additional resources

- CSI volume snapshots

3.2.9. DeploymentConfig capability

Purpose
The **DeploymentConfig** capability enables and manages the **DeploymentConfig** API.

> **IMPORTANT**
>
> If you disable the **DeploymentConfig** capability, the following resources will not be available in the cluster:
>
> - **DeploymentConfig** resources
> - The **deployer** service account
>
> Disable the **DeploymentConfig** capability only if you do not require **DeploymentConfig** resources and the **deployer** service account in the cluster.

3.2.10. Ingress Capability

Purpose
The Ingress Operator provides the features for the **Ingress** capability.

The Ingress Operator configures and manages the OpenShift Container Platform router.

Project
openshift-ingress-operator

CRDs

- **clusteringresses.ingress.openshift.io**
  - Scope: Namespaced
  - CR: **clusteringresses**
  - Validation: No

Configuration objects

- Cluster config
  - Type Name: **clusteringresses.ingress.openshift.io**
  - Instance Name: **default**
  - View Command:

    ```
    $ oc get clusteringresses.ingress.openshift.io -n openshift-ingress-operator default -o yaml
    ```

Notes
The Ingress Operator sets up the router in the **openshift-ingress** project and creates the deployment for the router:

```
$ oc get deployment -n openshift-ingress
```

The Ingress Operator uses the **clusterNetwork[].cidr** from the **network/cluster** status to determine what mode (IPv4, IPv6, or dual stack) the managed Ingress Controller (router) should operate in. For example, if **clusterNetwork** contains only a v6 **cidr**, then the Ingress Controller operates in IPv6-only mode.

In the following example, Ingress Controllers managed by the Ingress Operator will run in IPv4-only mode because only one cluster network exists and the network is an IPv4 **cidr**:

```
$ oc get network/cluster -o jsonpath='{.status.clusterNetwork[*]}'
```

Example output

```
map[cidr:10.128.0.0/14 hostPrefix:23]
```

3.2.11. Insights capability

Purpose
The Insights Operator provides the features for the **Insights** capability.

The Insights Operator gathers OpenShift Container Platform configuration data and sends it to Red Hat. The data is used to produce proactive insights recommendations about potential issues that a cluster might be exposed to. These insights are communicated to cluster administrators through the Insights advisor service on console.redhat.com.

Notes
Insights Operator complements OpenShift Container Platform Telemetry.

Additional resources

- Using Insights Operator

3.2.12. Machine API capability

Purpose
The **machine-api-operator**, **cluster-autoscaler-operator**, and **cluster-control-plane-machine-set-operator** Operators provide the features for the **MachineAPI** capability. You can disable this capability only if you install a cluster with user-provisioned infrastructure.

The Machine API capability is responsible for all machine configuration and management in the cluster. If you disable the Machine API capability during installation, you need to manage all machine-related tasks manually.

Additional resources

- Overview of machine management
- Machine API Operator
- Cluster Autoscaler Operator
- Control Plane Machine Set Operator

3.2.13. Marketplace capability

Purpose

The Marketplace Operator provides the features for the **marketplace** capability.

The Marketplace Operator simplifies the process for bringing off-cluster Operators to your cluster by using a set of default Operator Lifecycle Manager (OLM) catalogs on the cluster. When the Marketplace Operator is installed, it creates the **openshift-marketplace** namespace. OLM ensures catalog sources installed in the **openshift-marketplace** namespace are available for all namespaces on the cluster.

If you disable the **marketplace** capability, the Marketplace Operator does not create the **openshift-marketplace** namespace. Catalog sources can still be configured and managed on the cluster manually, but OLM depends on the **openshift-marketplace** namespace in order to make catalogs available to all namespaces on the cluster. Users with elevated permissions to create namespaces prefixed with **openshift-**, such as system or cluster administrators, can manually create the **openshift-marketplace** namespace.

If you enable the **marketplace** capability, you can enable and disable individual catalogs by configuring the Marketplace Operator.

Additional resources

- Red Hat-provided Operator catalogs

3.2.14. Node Tuning capability

Purpose

The Node Tuning Operator provides features for the **NodeTuning** capability.

The Node Tuning Operator helps you manage node-level tuning by orchestrating the TuneD daemon and achieves low latency performance by using the Performance Profile controller. The majority of high-performance applications require some level of kernel tuning. The Node Tuning Operator provides a unified management interface to users of node-level sysctls and more flexibility to add custom tuning specified by user needs.

If you disable the NodeTuning capability, some default tuning settings will not be applied to the control-plane nodes. This might limit the scalability and performance of large clusters with over 900 nodes or 900 routes.

Additional resources

- Using the Node Tuning Operator

3.2.15. OpenShift samples capability

Purpose

The Cluster Samples Operator provides the features for the **openshift-samples** capability.

The Cluster Samples Operator manages the sample image streams and templates stored in the **openshift** namespace.

On initial start up, the Operator creates the default samples configuration resource to initiate the creation of the image streams and templates. The configuration object is a cluster scoped object with the key **cluster** and type **configs.samples**.

The image streams are the Red Hat Enterprise Linux CoreOS (RHCOS)-based OpenShift Container Platform image streams pointing to images on **registry.redhat.io**. Similarly, the templates are those categorized as OpenShift Container Platform templates.

If you disable the samples capability, users cannot access the image streams, samples, and templates it provides. Depending on your deployment, you might want to disable this component if you do not need it.

Additional resources

- Configuring the Cluster Samples Operator

3.2.16. Operator Lifecycle Manager (OLM) Classic capability

Purpose
OLM (Classic) provides the features for the **OperatorLifecycleManager** capability.

Operator Lifecycle Manager (OLM) Classic helps users install, update, and manage the lifecycle of Kubernetes native applications (Operators) and their associated services running across their OpenShift Container Platform clusters. It is part of the Operator Framework, an open source toolkit designed to manage Operators in an effective, automated, and scalable way.

If an Operator requires any of the following APIs, then you must enable the **OperatorLifecycleManager** capability:

- **ClusterServiceVersion**
- **CatalogSource**
- **Subscription**
- **InstallPlan**
- **OperatorGroup**

IMPORTANT

The **marketplace** capability depends on the **OperatorLifecycleManager** capability. You cannot disable the **OperatorLifecycleManager** capability and enable the **marketplace** capability.

Additional resources

- Operator Lifecycle Manager concepts and resources

3.2.17. Operator Lifecycle Manager (OLM) v1 capability

Purpose
OLM v1 provides the features for the **OperatorLifecycleManagerV1** capability.

Starting in OpenShift Container Platform 4.18, OLM v1 is enabled by default alongside OLM (Classic). This next-generation iteration provides an updated framework that evolves many of OLM (Classic) concepts that enable cluster administrators to extend capabilities for their users.

OLM v1 manages the lifecycle of the new **ClusterExtension** object, which includes Operators via the **registry+v1** bundle format, and controls installation, upgrade, and role-based access control (RBAC) of extensions within a cluster.

In OpenShift Container Platform, OLM v1 is provided by the **olm** cluster Operator.

NOTE

The **olm** cluster Operator informs cluster administrators if there are any installed extensions blocking cluster upgrade, based on their **olm.maxOpenShiftVersion** properties. For more information, see "Compatibility with OpenShift Container Platform versions".

Components

Operator Lifecycle Manager (OLM) v1 comprises the following component projects:

Operator Controller

The central component of OLM v1 that extends Kubernetes with an API through which users can install and manage the lifecycle of Operators and extensions. It consumes information from catalogd.

Catalogd

A Kubernetes extension that unpacks file-based catalog (FBC) content packaged and shipped in container images for consumption by on-cluster clients. As a component of the OLM v1 microservices architecture, catalogd hosts metadata for Kubernetes extensions packaged by the authors of the extensions, and as a result helps users discover installable content.

CRDs

- **clusterextension.olm.operatorframework.io**
  - Scope: Cluster
  - CR: **ClusterExtension**
- **clustercatalog.olm.operatorframework.io**
  - Scope: Cluster
  - CR: **ClusterCatalog**

Project

- operator-framework/operator-controller
- operator-framework/catalogd

Additional resources

- Extensions overview

3.3. VIEWING THE CLUSTER CAPABILITIES

As a cluster administrator, you can view the capabilities by using the **clusterversion** resource status.

Prerequisites

- You have installed the OpenShift CLI (**oc**).

Procedure

- To view the status of the cluster capabilities, run the following command:

```
$ oc get clusterversion version -o jsonpath='{.spec.capabilities}{"\n"}{.status.capabilities}{"\n"}'
```

Example output

```
{"additionalEnabledCapabilities":["openshift-samples"],"baselineCapabilitySet":"None"}
{"enabledCapabilities":["openshift-samples"],"knownCapabilities":
["CSISnapshot","Console","Insights","Storage","baremetal","marketplace","openshift-
samples"]}
```

3.4. ENABLING THE CLUSTER CAPABILITIES BY SETTING BASELINE CAPABILITY SET

As a cluster administrator, you can enable cluster capabilities any time after a OpenShift Container Platform installation by setting the **baselineCapabilitySet** configuration parameter.

Prerequisites

- You have installed the OpenShift CLI (**oc**).

Procedure

- To set the **baselineCapabilitySet** configuration parameter, run the following command:

```
$ oc patch clusterversion version --type merge -p '{"spec":{"capabilities":
{"baselineCapabilitySet":"vCurrent"}}}' (1)
```

(1) For **baselineCapabilitySet** you can specify **vCurrent**, **v4.19**, or **None**.

3.5. ENABLING THE CLUSTER CAPABILITIES BY SETTING ADDITIONAL ENABLED CAPABILITIES

As a cluster administrator, you can enable cluster capabilities any time after a OpenShift Container Platform installation by setting the **additionalEnabledCapabilities** configuration parameter.

Prerequisites

- You have installed the OpenShift CLI (**oc**).

Procedure

1. View the additional enabled capabilities by running the following command:

```
$ oc get clusterversion version -o jsonpath='{.spec.capabilities.additionalEnabledCapabilities}
{"\n"}'
```

Example output

```
["openshift-samples"]
```

2. To set the **additionalEnabledCapabilities** configuration parameter, run the following command:

```
$ oc patch clusterversion/version --type merge -p '{"spec":{"capabilities":
{"additionalEnabledCapabilities":["openshift-samples", "marketplace"]}}}'
```

IMPORTANT

It is not possible to disable a capability which is already enabled in a cluster. The cluster version Operator (CVO) continues to reconcile the capability which is already enabled in the cluster.

If you try to disable a capability, then CVO shows the divergent spec:

```
$ oc get clusterversion version -o jsonpath='{.status.conditions[?
(@.type=="ImplicitlyEnabledCapabilities")]}{"\n"}'
```

Example output

```
{"lastTransitionTime":"2022-07-22T03:14:35Z","message":"The following capabilities could not be
disabled: openshift-
samples","reason":"CapabilitiesImplicitlyEnabled","status":"True","type":"ImplicitlyEnabledCapabilities"}
```

NOTE

During the cluster upgrades, it is possible that a given capability could be implicitly enabled. If a resource was already running on the cluster before the upgrade, then any capabilities that is part of the resource will be enabled. For example, during a cluster upgrade, a resource that is already running on the cluster has been changed to be part of the **marketplace** capability by the system. Even if a cluster administrator does not explicitly enabled the **marketplace** capability, it is implicitly enabled by the system.

CHAPTER 4. SUPPORT FOR FIPS CRYPTOGRAPHY

You can install an OpenShift Container Platform cluster in FIPS mode.

OpenShift Container Platform is designed for FIPS. When running Red Hat Enterprise Linux (RHEL) or Red Hat Enterprise Linux CoreOS (RHCOS) booted in FIPS mode, OpenShift Container Platform core components use the RHEL cryptographic libraries that have been submitted to NIST for FIPS 140-2/140-3 Validation on only the x86_64, ppc64le, and s390x architectures.

For more information about the NIST validation program, see Cryptographic Module Validation Program. For the latest NIST status for the individual versions of RHEL cryptographic libraries that have been submitted for validation, see Compliance Activities and Government Standards.

IMPORTANT

To enable FIPS mode for your cluster, you must run the installation program from a RHEL 9 computer that is configured to operate in FIPS mode, and you must use a FIPS-capable version of the installation program. See the section titled *Obtaining a FIPS-capable installation program using `oc adm extract`*.

For more information about configuring FIPS mode on RHEL, see Installing the system in FIPS mode.

For the Red Hat Enterprise Linux CoreOS (RHCOS) machines in your cluster, this change is applied when the machines are deployed based on the status of an option in the **install-config.yaml** file, which governs the cluster options that a user can change during cluster deployment. With Red Hat Enterprise Linux (RHEL) machines, you must enable FIPS mode when you install the operating system on the machines that you plan to use as worker machines.

Because FIPS must be enabled before the operating system that your cluster uses boots for the first time, you cannot enable FIPS after you deploy a cluster.

4.1. OBTAINING A FIPS-CAPABLE INSTALLATION PROGRAM USINGoc ADM EXTRACT

OpenShift Container Platform requires the use of a FIPS-capable installation binary to install a cluster in FIPS mode. You can obtain this binary by extracting it from the release image by using the OpenShift CLI (**oc**). After you have obtained the binary, you proceed with the cluster installation, replacing all instances of the **openshift-install** command with **openshift-install-fips**.

Prerequisites

- You have installed the OpenShift CLI (**oc**) with version 4.16 or newer.

Procedure

1. Extract the FIPS-capable binary from the installation program by running the following command:

```
$ oc adm release extract --registry-config "${pullsecret_file}" --command=openshift-install-fips --to "${extract_dir}" ${RELEASE_IMAGE}
```

   where:

<pullsecret_file>
: Specifies the name of a file that contains your pull secret.

<extract_dir>
: Specifies the directory where you want to extract the binary.

<RELEASE_IMAGE>
: Specifies the Quay.io URL of the OpenShift Container Platform release you are using. For more information on finding the release image, see *Extracting the OpenShift Container Platform installation program*.

2. Proceed with cluster installation, replacing all instances of the **openshift-install** command with **openshift-install-fips**.

Additional resources

- Extracting the OpenShift Container Platform installation program

4.2. OBTAINING A FIPS-CAPABLE INSTALLATION PROGRAM USING THE PUBLIC OPENSHIFT MIRROR

OpenShift Container Platform requires the use of a FIPS-capable installation binary to install a cluster in FIPS mode. You can obtain this binary by downloading it from the public OpenShift mirror. After you have obtained the binary, proceed with the cluster installation, replacing all instances of the **openshift-install** binary with **openshift-install-fips**.

Prerequisites

- You have access to the internet.

Procedure

1. Download the installation program from https://mirror.openshift.com/pub/openshift-v4/clients/ocp/latest-4.18/openshift-install-rhel9-amd64.tar.gz.

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

```
$ tar -xvf openshift-install-rhel9-amd64.tar.gz
```

3. Proceed with cluster installation, replacing all instances of the **openshift-install** command with **openshift-install-fips**.

4.3. FIPS VALIDATION IN OPENSHIFT CONTAINER PLATFORM

OpenShift Container Platform uses certain FIPS validated or Modules In Process modules within RHEL and RHCOS for the operating system components that it uses. See RHEL core crypto components. For example, when users use SSH to connect to OpenShift Container Platform clusters and containers, those connections are properly encrypted.

OpenShift Container Platform components are written in Go and built with Red Hat's golang compiler. When you enable FIPS mode for your cluster, all OpenShift Container Platform components that require cryptographic signing call RHEL and RHCOS cryptographic libraries.

Table 4.1. FIPS mode attributes and limitations in OpenShift Container Platform 4.19

Attributes	Limitations
FIPS support in RHEL 9 and RHCOS operating systems.	The FIPS implementation does not use a function that performs hash computation and signature generation or validation in a single step. This limitation will continue to be evaluated and improved in future OpenShift Container Platform releases.
FIPS support in CRI-O runtimes.	
FIPS support in OpenShift Container Platform services.	
FIPS validated or Modules In Process cryptographic module and algorithms that are obtained from RHEL 9 and RHCOS binaries and images.	
Use of FIPS compatible golang compiler.	TLS FIPS support is not complete but is planned for future OpenShift Container Platform releases.
FIPS support across multiple architectures.	FIPS is currently only supported on OpenShift Container Platform deployments using **x86_64**, **ppc64le**, and **s390x** architectures.

4.4. FIPS SUPPORT IN COMPONENTS THAT THE CLUSTER USES

Although the OpenShift Container Platform cluster itself uses FIPS validated or Modules In Process modules, ensure that the systems that support your OpenShift Container Platform cluster use FIPS validated or Modules In Process modules for cryptography.

4.4.1. etcd

To ensure that the secrets that are stored in etcd use FIPS validated or Modules In Process encryption, boot the node in FIPS mode. After you install the cluster in FIPS mode, you can encrypt the etcd data by using the FIPS-approved **aes cbc** cryptographic algorithm.

4.4.2. Storage

For local storage, use RHEL-provided disk encryption or Container Native Storage that uses RHEL-provided disk encryption. By storing all data in volumes that use RHEL-provided disk encryption and enabling FIPS mode for your cluster, both data at rest and data in motion, or network data, are protected by FIPS validated or Modules In Process encryption. You can configure your cluster to encrypt the root filesystem of each node, as described in Customizing nodes.

4.4.3. Runtimes

To ensure that containers know that they are running on a host that is using FIPS validated or Modules In Process cryptography modules, use CRI-O to manage your runtimes.

4.5. INSTALLING A CLUSTER IN FIPS MODE

To install a cluster in FIPS mode, follow the instructions to install a customized cluster on your preferred infrastructure. Ensure that you set **fips: true** in the **install-config.yaml** file before you deploy your cluster.

IMPORTANT

To enable FIPS mode for your cluster, you must run the installation program from a RHEL computer configured to operate in FIPS mode. For more information about configuring FIPS mode on RHEL, see Installing the system in FIPS mode .

- Amazon Web Services
- Microsoft Azure
- Bare metal
- Google Cloud Platform
- IBM Cloud®
- IBM Power®
- IBM Z® and IBM® LinuxONE
- IBM Z® and IBM® LinuxONE with RHEL KVM
- IBM Z® and IBM® LinuxONE in an LPAR
- Red Hat OpenStack Platform (RHOSP)
- VMware vSphere

NOTE

If you are using Azure File storage, you cannot enable FIPS mode.

To apply **AES CBC** encryption to your etcd data store, follow the Encrypting etcd data process after you install your cluster.