

Red Hat

OpenShift Container Platform 4.19

Installing on VMware vSphere

Installing OpenShift Container Platform on vSphere

OpenShift Container Platform 4.19 Installing on VMware vSphere

Installing OpenShift Container Platform on vSphere

Legal Notice

Copyright © Red Hat.

Except as otherwise noted below, the text of and illustrations in this documentation are licensed by Red Hat under the Creative Commons Attribution–Share Alike 3.0 Unported license . If you distribute this document or an adaptation of it, you must provide the URL for the original version.

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

Red Hat, the Red Hat logo, JBoss, Hibernate, and RHCE are trademarks or registered trademarks of Red Hat, LLC. or its subsidiaries in the United States and other countries.

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

XFS is a trademark or registered trademark of Hewlett Packard Enterprise Development LP or its subsidiaries in the United States and other countries.

The OpenStack ® Word Mark and OpenStack logo are trademarks or registered trademarks of the Linux Foundation, used under license.

All other trademarks are the property of their respective owners.

Abstract

This document describes how to install OpenShift Container Platform on vSphere.

Table of Contents

CHAPTER 1. INSTALLATION METHODS	8
1.1. ASSISTED INSTALLER	8
1.2. AGENT-BASED INSTALLER	8
1.3. INSTALLER-PROVISIONED INFRASTRUCTURE INSTALLATION	8
1.4. USER-PROVISIONED INFRASTRUCTURE INSTALLATION	8
1.5. ADDITIONAL RESOURCES	9
CHAPTER 2. INSTALLER-PROVISIONED INFRASTRUCTURE	10
2.1. VSphere INSTALLATION REQUIREMENTS	10
2.1.1. VMware vSphere infrastructure requirements	10
2.1.2. Network connectivity requirements	12
2.1.3. VMware vSphere CSI Driver Operator requirements	13
2.1.4. vCenter requirements	14
2.1.4.1. Required vCenter account privileges	14
2.1.4.2. Minimum required vCenter account privileges	23
2.1.4.3. Using OpenShift Container Platform with vMotion	31
2.1.4.4. Cluster resources	32
2.1.4.5. Cluster limits	33
2.1.4.6. Networking requirements	33
2.1.4.6.1. Required IP addresses	33
2.1.4.6.2. DNS records	33
2.1.4.6.3. Static IP addresses for vSphere nodes	34
2.2. PREPARING TO INSTALL A CLUSTER USING INSTALLER-PROVISIONED INFRASTRUCTURE	35
2.2.1. Obtaining the installation program	36
2.2.2. Installing the OpenShift CLI on Linux	37
2.2.3. Installing the OpenShift CLI on Windows	38
2.2.4. Installing the OpenShift CLI on macOS	38
2.2.5. Generating a key pair for cluster node SSH access	39
2.2.6. Adding vCenter root CA certificates to your system trust	41
2.3. INSTALLING A CLUSTER ON VSphere	42
2.3.1. Prerequisites	42
2.3.2. Internet access for OpenShift Container Platform	42
2.3.3. Deploying the cluster	43
2.3.4. Logging in to the cluster by using the CLI	46
2.3.5. Creating registry storage	46
2.3.5.1. Image registry removed during installation	47
2.3.5.2. Image registry storage configuration	47
2.3.5.2.1. Configuring registry storage for VMware vSphere	47
2.3.5.2.2. Configuring block registry storage for VMware vSphere	49
2.3.5.6. Telemetry access for OpenShift Container Platform	50
2.3.7. Next steps	50
2.4. INSTALLING A CLUSTER ON VSphere WITH CUSTOMIZATIONS	51
2.4.1. Prerequisites	51
2.4.2. Internet access for OpenShift Container Platform	51
2.4.3. VMware vSphere region and zone enablement	52
2.4.4. VMware vSphere host group enablement	53
2.4.5. Creating the installation configuration file	54
2.4.5.1. Sample install-config.yaml file for an installer-provisioned VMware vSphere cluster	57
2.4.5.2. Configuring the cluster-wide proxy during installation	59
2.4.5.3. Configuring regions and zones for a VMware vCenter	61
2.4.5.4. Configuring host groups for a VMware vCenter	63

2.4.6. Services for a user-managed load balancer	66
2.4.6.1. Configuring a user-managed load balancer	68
2.4.7. Deploying the cluster	75
2.4.8. Logging in to the cluster by using the CLI	77
2.4.9. Creating registry storage	78
2.4.9.1. Image registry removed during installation	78
2.4.9.2. Image registry storage configuration	78
2.4.9.2.1. Configuring registry storage for VMware vSphere	78
2.4.9.2.2. Configuring block registry storage for VMware vSphere	80
2.4.10. Telemetry access for OpenShift Container Platform	81
2.4.11. Next steps	81
2.5. INSTALLING A CLUSTER ON VSphere WITH NETWORK CUSTOMIZATIONS	82
2.5.1. Prerequisites	82
2.5.2. Internet access for OpenShift Container Platform	82
2.5.3. VMware vSphere region and zone enablement	83
2.5.4. VMware vSphere host group enablement	84
2.5.5. Creating the installation configuration file	85
2.5.5.1. Sample <code>install-config.yaml</code> file for an installer-provisioned VMware vSphere cluster	88
2.5.5.2. Configuring the cluster-wide proxy during installation	90
2.5.5.3. Deploying with dual-stack networking	92
2.5.5.4. Configuring regions and zones for a VMware vCenter	93
2.5.5.5. Configuring host groups for a VMware vCenter	95
2.5.5.6. Configuring multiple NICs	97
2.5.6. Network configuration phases	99
2.5.7. Specifying advanced network configuration	100
2.5.7.1. Specifying multiple subnets for your network	101
2.5.8. Cluster Network Operator configuration	102
2.5.8.1. Cluster Network Operator configuration object	103
2.5.8.2. <code>defaultNetwork</code> object configuration	104
2.5.8.3. Configuration for the OVN-Kubernetes network plugin	104
2.5.9. Services for a user-managed load balancer	110
2.5.9.1. Configuring a user-managed load balancer	113
2.5.10. Deploying the cluster	120
2.5.11. Logging in to the cluster by using the CLI	122
2.5.12. Creating registry storage	122
2.5.12.1. Image registry removed during installation	122
2.5.12.2. Image registry storage configuration	122
2.5.12.2.1. Configuring registry storage for VMware vSphere	123
2.5.12.2.2. Configuring block registry storage for VMware vSphere	124
2.5.13. Telemetry access for OpenShift Container Platform	126
2.5.14. Configuring network components to run on the control plane	126
2.5.15. Next steps	128
2.6. INSTALLING A CLUSTER ON VSphere IN A DISCONNECTED ENVIRONMENT	128
2.6.1. Prerequisites	129
2.6.2. About installations in restricted networks	129
2.6.2.1. Additional limits	130
2.6.3. Internet access for OpenShift Container Platform	130
2.6.4. Creating the RHCOS image for restricted network installations	130
2.6.5. VMware vSphere region and zone enablement	131
2.6.6. VMware vSphere host group enablement	132
2.6.7. Creating the installation configuration file	133
2.6.7.1. Sample <code>install-config.yaml</code> file for an installer-provisioned VMware vSphere cluster	137
2.6.7.2. Configuring the cluster-wide proxy during installation	140

2.6.7.3. Configuring regions and zones for a VMware vCenter	142
2.6.7.4. Configuring host groups for a VMware vCenter	144
2.6.8. Services for a user-managed load balancer	146
2.6.8.1. Configuring a user-managed load balancer	149
2.6.9. Deploying the cluster	156
2.6.10. Logging in to the cluster by using the CLI	158
2.6.11. Disabling the default OperatorHub catalog sources	159
2.6.12. Creating registry storage	159
2.6.12.1. Image registry removed during installation	159
2.6.12.2. Image registry storage configuration	159
2.6.12.2.1. Configuring registry storage for VMware vSphere	159
2.6.13. Telemetry access for OpenShift Container Platform	161
2.6.14. Next steps	161
CHAPTER 3. USER-PROVISIONED INFRASTRUCTURE	162
3.1. VSPHERE INSTALLATION REQUIREMENTS FOR USER-PROVISIONED INFRASTRUCTURE	162
3.1.1. VMware vSphere infrastructure requirements	162
3.1.2. VMware vSphere CSI Driver Operator requirements	164
3.1.3. Requirements for a cluster with user-provisioned infrastructure	165
3.1.3.1. vCenter requirements	165
3.1.3.1.1. Required vCenter account privileges	165
3.1.3.1.2. Minimum required vCenter account privileges	173
3.1.3.1.3. Using OpenShift Container Platform with vMotion	178
3.1.3.1.4. Cluster resources	179
3.1.3.1.5. Cluster limits	180
3.1.3.1.6. Networking requirements	180
3.1.3.1.6.1. DNS records	180
3.1.3.2. Required machines for cluster installation	181
3.1.3.3. Minimum resource requirements for cluster installation	182
3.1.3.4. Requirements for encrypting virtual machines	183
3.1.3.5. Certificate signing requests management	184
3.1.3.6. Networking requirements for user-provisioned infrastructure	184
3.1.3.6.1. Setting the cluster node hostnames through DHCP	185
3.1.3.6.2. Network connectivity requirements	185
3.1.3.6.3. NTP configuration for user-provisioned infrastructure	186
3.1.3.7. User-provisioned DNS requirements	186
3.1.3.7.1. Example DNS configuration for user-provisioned clusters	188
3.1.3.8. Load balancing requirements for user-provisioned infrastructure	191
3.1.3.8.1. Example load balancer configuration for user-provisioned clusters	192
3.2. PREPARING TO INSTALL A CLUSTER USING USER-PROVISIONED INFRASTRUCTURE	194
3.2.1. Obtaining the installation program	195
3.2.2. Installing the OpenShift CLI on Linux	196
3.2.3. Installing the OpenShift CLI on Windows	197
3.2.4. Installing the OpenShift CLI on macOS	197
3.2.5. Generating a key pair for cluster node SSH access	198
3.2.6. Preparing the user-provisioned infrastructure	200
3.2.6.1. Example load balancer configuration for user-provisioned clusters	202
3.2.7. Validating DNS resolution for user-provisioned infrastructure	204
3.3. INSTALLING A CLUSTER ON VSPHERE WITH USER-PROVISIONED INFRASTRUCTURE	206
3.3.1. Prerequisites	206
3.3.2. Internet access for OpenShift Container Platform	207
3.3.3. VMware vSphere region and zone enablement	207
3.3.4. Manually creating the installation configuration file	209

3.3.4.1. Sample <code>install-config.yaml</code> file for VMware vSphere	210
3.3.4.2. Configuring the cluster-wide proxy during installation	213
3.3.4.3. Configuring regions and zones for a VMware vCenter	215
3.3.5. Creating the Kubernetes manifest and Ignition config files	217
3.3.6. Extracting the infrastructure name	219
3.3.7. Installing RHCOS and starting the OpenShift Container Platform bootstrap process	219
3.3.8. Adding more compute machines to a cluster in vSphere	224
3.3.9. Disk partitioning	225
3.3.10. Creating a separate <code>/var</code> partition	226
3.3.11. Waiting for the bootstrap process to complete	228
3.3.12. Logging in to the cluster by using the CLI	229
3.3.13. Approving the certificate signing requests for your machines	229
3.3.14. Initial Operator configuration	232
3.3.14.1. Image registry removed during installation	233
3.3.14.2. Image registry storage configuration	233
3.3.14.2.1. Configuring registry storage for VMware vSphere	234
3.3.14.2.2. Configuring storage for the image registry in non-production clusters	235
3.3.14.2.3. Configuring block registry storage for VMware vSphere	236
3.3.15. Completing installation on user-provisioned infrastructure	237
3.3.16. Configuring vSphere DRS anti-affinity rules for control plane nodes	240
3.3.17. Telemetry access for OpenShift Container Platform	241
3.3.18. Next steps	241
3.4. INSTALLING A CLUSTER ON VSphere WITH NETWORK CUSTOMIZATIONS	241
3.4.1. Prerequisites	242
3.4.2. Internet access for OpenShift Container Platform	242
3.4.3. VMware vSphere region and zone enablement	243
3.4.4. Manually creating the installation configuration file	244
3.4.4.1. Sample <code>install-config.yaml</code> file for VMware vSphere	245
3.4.4.2. Configuring the cluster-wide proxy during installation	248
3.4.4.3. Configuring regions and zones for a VMware vCenter	250
3.4.5. Network configuration phases	252
3.4.6. Specifying advanced network configuration	253
3.4.6.1. Specifying multiple subnets for your network	254
3.4.7. Cluster Network Operator configuration	256
3.4.7.1. Cluster Network Operator configuration object	256
3.4.7.2. <code>defaultNetwork</code> object configuration	257
3.4.7.3. Configuration for the OVN-Kubernetes network plugin	258
3.4.8. Creating the Ignition config files	263
3.4.9. Extracting the infrastructure name	264
3.4.10. Installing RHCOS and starting the OpenShift Container Platform bootstrap process	265
3.4.11. Adding more compute machines to a cluster in vSphere	269
3.4.12. Disk partitioning	271
3.4.13. Creating a separate <code>/var</code> partition	271
3.4.14. Waiting for the bootstrap process to complete	273
3.4.15. Logging in to the cluster by using the CLI	274
3.4.16. Approving the certificate signing requests for your machines	275
3.4.16.1. Initial Operator configuration	278
3.4.16.2. Image registry removed during installation	279
3.4.16.3. Image registry storage configuration	279
3.4.16.3.1. Configuring block registry storage for VMware vSphere	279
3.4.17. Completing installation on user-provisioned infrastructure	280
3.4.18. Configuring vSphere DRS anti-affinity rules for control plane nodes	283
3.4.19. Telemetry access for OpenShift Container Platform	284

3.4.20. Next steps	284
3.5. INSTALLING A CLUSTER ON VSphere IN A DISCONNECTED ENVIRONMENT WITH USER-PROVISIONED INFRASTRUCTURE	285
3.5.1. Prerequisites	285
3.5.2. About installations in restricted networks	286
3.5.2.1. Additional limits	286
3.5.3. Internet access for OpenShift Container Platform	286
3.5.4. VMware vSphere region and zone enablement	286
3.5.5. Manually creating the installation configuration file	288
3.5.5.1. Sample install-config.yaml file for VMware vSphere	290
3.5.5.2. Configuring the cluster-wide proxy during installation	293
3.5.5.3. Configuring regions and zones for a VMware vCenter	295
3.5.6. Creating the Kubernetes manifest and Ignition config files	297
3.5.7. Configuring chrony time service	298
3.5.8. Extracting the infrastructure name	300
3.5.9. Installing RHCOS and starting the OpenShift Container Platform bootstrap process	300
3.5.10. Adding more compute machines to a cluster in vSphere	305
3.5.11. Disk partitioning	306
3.5.12. Creating a separate /var partition	307
3.5.13. Waiting for the bootstrap process to complete	309
3.5.14. Logging in to the cluster by using the CLI	310
3.5.15. Approving the certificate signing requests for your machines	310
3.5.16. Initial Operator configuration	313
3.5.16.1. Disabling the default OperatorHub catalog sources	314
3.5.16.2. Image registry storage configuration	314
3.5.16.2.1. Configuring registry storage for VMware vSphere	315
3.5.16.2.2. Configuring storage for the image registry in non-production clusters	316
3.5.16.2.3. Configuring block registry storage for VMware vSphere	317
3.5.17. Completing installation on user-provisioned infrastructure	318
3.5.18. Configuring vSphere DRS anti-affinity rules for control plane nodes	321
3.5.19. Telemetry access for OpenShift Container Platform	322
3.5.20. Next steps	322
CHAPTER 4. INSTALLING A CLUSTER ON VSphere USING THE ASSISTED INSTALLER	323
4.1. ADDITIONAL RESOURCES	323
CHAPTER 5. INSTALLING A CLUSTER ON VSphere USING THE AGENT-BASED INSTALLER	324
CHAPTER 6. INSTALLING A THREE-NODE CLUSTER ON VSphere	325
6.1. CONFIGURING A THREE-NODE CLUSTER	325
6.2. NEXT STEPS	326
CHAPTER 7. UNINSTALLING A CLUSTER ON VSphere THAT USES INSTALLER-PROVISIONED INFRASTRUCTURE	327
7.1. REMOVING A CLUSTER THAT USES INSTALLER-PROVISIONED INFRASTRUCTURE	327
CHAPTER 8. USING THE VSphere PROBLEM DETECTOR OPERATOR	328
8.1. ABOUT THE VSphere PROBLEM DETECTOR OPERATOR	328
8.2. RUNNING THE VSphere PROBLEM DETECTOR OPERATOR CHECKS	328
8.3. VIEWING THE EVENTS FROM THE VSphere PROBLEM DETECTOR OPERATOR	329
8.4. VIEWING THE LOGS FROM THE VSphere PROBLEM DETECTOR OPERATOR	330
8.5. CONFIGURATION CHECKS RUN BY THE VSphere PROBLEM DETECTOR OPERATOR	330
8.6. ABOUT THE STORAGE CLASS CONFIGURATION CHECK	332
8.7. METRICS FOR THE VSphere PROBLEM DETECTOR OPERATOR	332

8.8. ADDITIONAL RESOURCES	333
CHAPTER 9. INSTALLATION CONFIGURATION PARAMETERS FOR VSphere	334
9.1. AVAILABLE INSTALLATION CONFIGURATION PARAMETERS FOR VSphere	334
9.1.1. Required configuration parameters	334
9.1.2. Network configuration parameters	335
9.1.3. Optional configuration parameters	339
9.1.4. Additional VMware vSphere configuration parameters	344
9.1.5. Deprecated VMware vSphere configuration parameters	352
9.1.6. Optional VMware vSphere machine pool configuration parameters	355
CHAPTER 10. MULTIPLE REGIONS AND ZONES CONFIGURATION FOR A CLUSTER ON VMWARE VSphere	357
10.1. SPECIFYING MULTIPLE REGIONS AND ZONES FOR YOUR CLUSTER ON VSphere	357
10.2. ENABLING A MULTIPLE LAYER 2 NETWORK FOR YOUR CLUSTER	359
10.3. PARAMETERS FOR THE CLUSTER-WIDE INFRASTRUCTURE CRD	360
10.4. SPECIFYING MULTIPLE HOST GROUPS FOR YOUR CLUSTER ON VSphere	361
CHAPTER 11. ENABLING ENCRYPTION ON A VSphere CLUSTER	366
11.1. ENCRYPTING VIRTUAL MACHINES	366
11.2. ADDITIONAL RESOURCES	366
CHAPTER 12. CONFIGURING THE VSphere CONNECTION SETTINGS AFTER AN INSTALLATION	368
12.1. CONFIGURING THE VSphere CONNECTION SETTINGS	368
12.2. VERIFYING THE CONFIGURATION	369

CHAPTER 1. INSTALLATION METHODS

You can install an OpenShift Container Platform cluster on vSphere using a variety of different installation methods. Each method has qualities that can make them more suitable for different use cases, such as installing a cluster in a disconnected environment or installing a cluster with minimal configuration and provisioning.

1.1. ASSISTED INSTALLER

You can install OpenShift Container Platform with the [Assisted Installer](#). This method requires no setup for the installer and is ideal for connected environments like vSphere. Installing with the Assisted Installer also provides integration with vSphere, enabling autoscaling. See [Installing an on-premise cluster using the Assisted Installer](#) for additional details.

1.2. AGENT-BASED INSTALLER

You can install an OpenShift Container Platform cluster on vSphere using the Agent-based Installer. The Agent-based Installer can be used to boot an on-premise server in a disconnected environment by using a bootable image. With the Agent-based Installer, users also have the flexibility to provision infrastructure, customize network configurations, and customize installations within a disconnected environment. See [Preparing to install with the Agent-based Installer](#) for additional details.

1.3. INSTALLER-PROVISIONED INFRASTRUCTURE INSTALLATION

You can install OpenShift Container Platform on vSphere by using installer-provisioned infrastructure. Installer-provisioned infrastructure allows the installation program to preconfigure and automate the provisioning of resources required by OpenShift Container Platform. Installer-provisioned infrastructure is useful for installing in environments with disconnected networks, where the installation program provisions the underlying infrastructure for the cluster.

- [Installing a cluster on vSphere](#) You can install OpenShift Container Platform on vSphere by using installer-provisioned infrastructure installation with no customization.
- [Installing a cluster on vSphere with customizations](#) You can install OpenShift Container Platform on vSphere by using installer-provisioned infrastructure installation with the default customization options.
- [Installing a cluster on vSphere with network customizations](#) You can install OpenShift Container Platform on installer-provisioned vSphere infrastructure, with network customizations. You can customize your OpenShift Container Platform network configuration during installation, so that your cluster can coexist with your existing IP address allocations and adhere to your network requirements.
- [Installing a cluster on vSphere in a restricted network](#) You can install a cluster on VMware vSphere infrastructure in a restricted network by creating an internal mirror of the installation release content. You can use this method to deploy OpenShift Container Platform on an internal network that is not visible to the internet.

1.4. USER-PROVISIONED INFRASTRUCTURE INSTALLATION

You can install OpenShift Container Platform on vSphere by using user-provisioned infrastructure. User-provisioned infrastructure requires the user to provision all resources required by OpenShift Container Platform. If you do not use infrastructure that the installation program provisions, you must manage and maintain the cluster resources yourself.

- **Installing a cluster on vSphere with user-provisioned infrastructure** You can install OpenShift Container Platform on VMware vSphere infrastructure that you provision.
- **Installing a cluster on vSphere with network customizations with user-provisioned infrastructure:** You can install OpenShift Container Platform on VMware vSphere infrastructure that you provision with customized network configuration options.
- **Installing a cluster on vSphere in a restricted network with user-provisioned infrastructure** OpenShift Container Platform can be installed on VMware vSphere infrastructure that you provision in a restricted network.



IMPORTANT

The steps for performing a user-provisioned infrastructure installation are provided as an example only. Installing a cluster with infrastructure you provide requires knowledge of the vSphere platform and the installation process of OpenShift Container Platform. Use the user-provisioned infrastructure installation instructions as a guide; you are free to create the required resources through other methods.

1.5. ADDITIONAL RESOURCES

- [Installation process](#)

CHAPTER 2. INSTALLER-PROVISIONED INFRASTRUCTURE

2.1. VSphere Installation Requirements

Before you begin an installation using installer-provisioned infrastructure, be sure that your vSphere environment meets the following installation requirements.

2.1.1. VMware vSphere infrastructure requirements

You must install an OpenShift Container Platform cluster on one of the following versions of a VMware vSphere instance that meets the requirements for the components that you use:

- Version 8.0 Update 1 or later, or VMware Cloud Foundation 5.0 or later
- VMware vSphere Foundation 9 or later, or VMware Cloud Foundation 9 or later

Both of these releases support Container Storage Interface (CSI) migration, which is enabled by default on OpenShift Container Platform 4.19.

You can host the VMware vSphere infrastructure on-premise or on a [VMware Cloud Verified provider](#) that meets the requirements outlined in the following tables:

Table 2.1. Version requirements for vSphere virtual environments

Virtual environment product	Required version
VMware virtual hardware	15 or later
vSphere ESXi hosts	8.0 Update 1 or later, or VMware vSphere Foundation 9 or later; VMware Cloud Foundation 5.0 or later, or VMware Cloud Foundation 9 or later
vCenter host	8.0 Update 1 or later, or VMware vSphere Foundation 9 or later; VMware Cloud Foundation 5.0 or later, or VMware Cloud Foundation 9 or later



IMPORTANT

You must ensure that the time on your ESXi hosts is synchronized before you install OpenShift Container Platform. See [Edit Time Configuration for a Host](#) in the VMware documentation.

Table 2.2. Minimum supported vSphere version for VMware components

Component	Minimum supported versions	Description
-----------	----------------------------	-------------

Component	Minimum supported versions	Description
Hypervisor	vSphere 8.0 Update 1 or later, or VMware Cloud Foundation 5.0 or later with virtual hardware version 15; VMware vSphere Foundation 9 or later, or VMware Cloud Foundation 9 or later	This hypervisor version is the minimum version that Red Hat Enterprise Linux CoreOS (RHCOS) supports. For more information about supported hardware on the latest version of Red Hat Enterprise Linux (RHEL) that is compatible with RHCOS, see Hardware on the Red Hat Customer Portal.
Networking (NSX)	vSphere 8.0 Update 1 or later, or VMware Cloud Foundation 5.0 or later; VMware vSphere Foundation 9 or later, or VMware Cloud Foundation 9 or later	Red Hat uses the Partner Certification process to verify NSX compatibility.
CPU micro-architecture	x86-64-v2 or higher	OpenShift Container Platform version 4.13 and later are based on the RHEL 9.2 host operating system, which raised the microarchitecture requirements to x86-64-v2. See Architectures in the RHEL documentation.



IMPORTANT

To ensure the best performance conditions for your cluster workloads that operate on Oracle® Cloud Infrastructure (OCI) and on the Oracle® Cloud VMware Solution (OCVS) service, ensure volume performance units (VPUs) for your block volume are sized for your workloads.

The following list provides some guidance in selecting the VPUs needed for specific performance needs:

- Test or proof of concept environment: 100 GB, and 20 to 30 VPUs.
- Base-production environment: 500 GB, and 60 VPUs.
- Heavy-use production environment: More than 500 GB, and 100 or more VPUs.

Consider allocating additional VPUs to give enough capacity for updates and scaling activities. See [Block Volume Performance Levels \(Oracle documentation\)](#).



NOTE

The following additional VMware vSphere Foundation and VMware Cloud Foundation components are outside the scope of Red Hat support:

- Management: VCF Operations, VCF Automation, VCF Fleet Management, and VCF Identity Broker.
- Networking: VMware NSX Container Plugin (NCP).
- Migration: VMware HCX.

2.1.2. Network connectivity requirements

You must configure the network connectivity between machines to allow OpenShift Container Platform cluster components to communicate.

Review the following details about the required network ports.

Table 2.3. Ports used for all-machine to all-machine communications

Protocol	Port	Description
VRRP	N/A	Required for keepalived
ICMP	N/A	Network reachability tests
TCP	1936	Metrics
	9000-9999	Host level services, including the node exporter on ports 9100-9101 and the Cluster Version Operator on port 9099 .
	10250-10259	The default ports that Kubernetes reserves
UDP	6081	Geneve
	9000-9999	Host level services, including the node exporter on ports 9100-9101 .
	500	IPsec IKE packets
	4500	IPsec NAT-T packets
	TCP/UDP	30000-32767
Kubernetes node port	ESP	N/A

Table 2.4. Ports used for all-machine to control plane communications

Protocol	Port	Description
TCP	6443	Kubernetes API

Table 2.5. Ports used for control plane machine to control plane machine communications

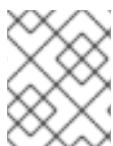
Protocol	Port	Description
TCP	2379-2380	etcd server and peer ports

2.1.3. VMware vSphere CSI Driver Operator requirements

To install the vSphere Container Storage Interface (CSI) Driver Operator, the following requirements must be met:

- VMware vSphere version 8.0 Update 1 or later; or VMware vSphere Foundation (VVF) 9; or VMware Cloud Foundation (VCF) 5 or later
- vCenter version 8.0 Update 1 or later; or VVF 9; or VCF 5 or later
- Virtual machines of hardware version 15 or later
- No third-party vSphere CSI driver already installed in the cluster

If a third-party vSphere CSI driver is present in the cluster, OpenShift Container Platform does not overwrite it. The presence of a third-party vSphere CSI driver prevents OpenShift Container Platform from updating to OpenShift Container Platform 4.13 or later.



NOTE

The VMware vSphere CSI Driver Operator is supported only on clusters deployed with **platform: vsphere** in the installation manifest.

You can create a custom role for the Container Storage Interface (CSI) driver, the vSphere CSI Driver Operator, and the vSphere Problem Detector Operator. The custom role can include privilege sets that assign a minimum set of permissions to each vSphere object. This means that the CSI driver, the vSphere CSI Driver Operator, and the vSphere Problem Detector Operator can establish a basic interaction with these objects.



IMPORTANT

Installing an OpenShift Container Platform cluster in a vCenter is tested against a full list of privileges as described in the "Required vCenter account privileges" section. By adhering to the full list of privileges, you can reduce the possibility of unexpected and unsupported behaviors that might occur when creating a custom role with a set of restricted privileges.

Additional resources

- To remove a third-party vSphere CSI driver, see [Removing a third-party vSphere CSI Driver](#) .

- To update the hardware version for your vSphere nodes, see [Updating hardware on nodes running in vSphere](#).
- [Minimum permissions for the storage components](#)

2.1.4. vCenter requirements

Before you install an OpenShift Container Platform cluster on your vCenter that uses infrastructure that the installation program provisions, you must prepare your environment.

2.1.4.1. Required vCenter account privileges

To install an OpenShift Container Platform cluster in a vCenter, the installation program requires access to an account with privileges to read and create the required resources. Using an account that has global administrative privileges is the simplest way to access all of the necessary permissions.

If you cannot use an account with global administrative privileges, you must create roles to grant the privileges necessary for OpenShift Container Platform cluster installation. Most of the privileges are always required. Some privileges are required only if you plan for the installation program to provision a folder to contain the OpenShift Container Platform cluster on your vCenter instance, which is the default behavior. You must create or change vSphere roles for the specified objects to grant the required privileges.

The installation program requires an additional role to create a vSphere virtual machine folder.

Example 2.1. Roles and privileges required for installation in vSphere API

vSphere object for role	When required	Required privileges in vSphere API
vSphere vCenter	Always	Cns.Searchable InventoryService.Tagging.AttachTag InventoryService.Tagging.CreateCategory InventoryService.Tagging.CreateTag InventoryService.Tagging.DeleteCategory InventoryService.Tagging.DeleteTag InventoryService.Tagging.EditCategory InventoryService.Tagging.EditTag Sessions.ValidateSession StorageProfile.Update StorageProfile.View

vSphere object for role	When required	Required privileges in vSphere API
vSphere vCenter Cluster	If VMs need creation in the cluster root	Host.Config.Storage Resource.AssignVMToPool VApp.AssignResourcePool VApp.Import VirtualMachine.Config.AddNewDisk
vSphere vCenter Resource Pool	For a provided existing resource pool	Resource.AssignVMToPool VApp.AssignResourcePool VApp.Import VirtualMachine.Config.AddNewDisk
vSphere Datastore	Always	Datastore.AllocateSpace Datastore.Browse Datastore.FileManagement InventoryService.Tagging.ObjectAttachable
vSphere Port Group	Always	Network.Assign
Virtual Machine Folder	Always	InventoryService.Tagging.ObjectAttachable Resource.AssignVMToPool VApp.Import VirtualMachine.Config.AddExistingDisk VirtualMachine.Config.AddNewDisk VirtualMachine.Config.AddRemoveDevice VirtualMachine.Config.AdvancedConfig VirtualMachine.Config.Annotation VirtualMachine.Config.CPUCount VirtualMachine.Config.DiskExtend VirtualMachine.Config.DiskLease VirtualMachine.Config.EditDevice VirtualMachine.Config.Memory VirtualMachine.Config.Rem

vSphere object for role	When required	Required privileges in vSphere API
		oveDisk VirtualMachine.Config.Remove Host.Config.Storage VirtualMachine.Config.RestGuestInfo VirtualMachine.Config.Resource VirtualMachine.Config.Settings VirtualMachine.Config.UpgradeVirtualHardware VirtualMachine.Interact.GuestControl VirtualMachine.Interact.PowerOff VirtualMachine.Interact.PowerOn VirtualMachine.Interact.Rest VirtualMachine.Inventory.Create VirtualMachine.Inventory.CreateFromExisting VirtualMachine.Inventory.Delete VirtualMachine.Provisioning.Clone VirtualMachine.Provisioning.MarkAsTemplate VirtualMachine.Provisioning.DeployTemplate
vSphere vCenter data center	The installation program creates the virtual machine folder.	InventoryService.Tagging.ObjectAttachable Resource.AssignVMTToPool VirtualMachine.Config.AddExistingDisk VirtualMachine.Config.AddNewDisk VirtualMachine.Config.AddRemoveDevice VirtualMachine.Config.AdvancedConfig VirtualMachine.Config.Annotation VirtualMachine.Config.CPUCount VirtualMachine.Config.DiskExtend VirtualMachine.Config.DiskLease

vSphere object for role	When required	Required privileges in vSphere API
		VirtualMachine.Config.Edit VirtualMachine.Config.RemoveDisk VirtualMachine.Config.Rename VirtualMachine.Config.RestGuestInfo VirtualMachine.Config.Resource VirtualMachine.Config.Settings VirtualMachine.Config.UpgradeVirtualHardware VirtualMachine.Interact.GuestControl VirtualMachine.Interact.PowerOff VirtualMachine.Interact.PowerOn VirtualMachine.Interact.Rest VirtualMachine.Inventory.Create VirtualMachine.Inventory.CreateFromExisting VirtualMachine.Inventory.Delete VirtualMachine.Provisioning.Clone VirtualMachine.Provisioning.DeployTemplate VirtualMachine.Provisioning.MarkAsTemplate Folder.Create Folder.Delete

Example 2.2. Roles and privileges required for installation in vCenter graphical user interface (GUI)

vSphere object for role	When required	Required privileges in vCenter GUI

vSphere object for role	When required	Required privileges in vCenter GUI
vSphere vCenter	Always	Cns.Searchable "vSphere Tagging"."Assign or Unassign vSphere Tag" "vSphere Tagging"."Create vSphere Tag Category" "vSphere Tagging"."Create vSphere Tag" "vSphere Tagging"."Delete vSphere Tag Category" "vSphere Tagging"."Delete vSphere Tag" "vSphere Tagging"."Edit vSphere Tag Category" "vSphere Tagging"."Edit vSphere Tag" Sessions."Validate session" "VM storage policies"."Update VM storage policies" "VM storage policies"."View VM storage policies"
vSphere vCenter Cluster	For VMs creation in the cluster root	Host.Configuration."Storage partition configuration" Resource."Assign virtual machine to resource pool" VApp."Assign resource pool" VApp.Import "Virtual machine"."Change Configuration"."Add new disk"
vSphere vCenter Resource Pool	If providing an existing resource pool	Host.Configuration."Storage partition configuration" Resource."Assign virtual machine to resource pool" VApp."Assign resource pool" VApp.Import "Virtual machine"."Change Configuration"."Add new disk"
vSphere Datastore	Always	Datastore."Allocate space" Datastore."Browse datastore" Datastore."Low level file operations" "vSphere Tagging"."Assign or Unassign vSphere Tag on Object"

vSphere object for role	When required	Required privileges in vCenter GUI
vSphere Port Group	Always	Network . "Assign network"
Virtual Machine Folder	Always	"vSphere Tagging" . "Assign or Unassign vSphere Tag on Object" Resource . "Assign virtual machine to resource pool" VApp.Import "Virtual machine" . "Change Configuration" . "Add existing disk" "Virtual machine" . "Change Configuration" . "Add new disk" "Virtual machine" . "Change Configuration" . "Add or remove device" "Virtual machine" . "Change Configuration" . "Advanced configuration" "Virtual machine" . "Change Configuration" . "Set annotation" "Virtual machine" . "Change Configuration" . "Change CPU count" "Virtual machine" . "Change Configuration" . "Extend virtual disk" "Virtual machine" . "Change Configuration" . "Acquire disk lease" "Virtual machine" . "Change Configuration" . "Modify device settings" "Virtual machine" . "Change Configuration" . "Change Memory" "Virtual machine" . "Change Configuration" . "Remove disk" "Virtual machine" . "Change Configuration" . "Rename" "Virtual machine" . "Change Configuration" . "Reset guest information" "Virtual machine" . "Change Configuration" . "Change resource" "Virtual machine" . "Change Configuration" . "Change"

vSphere object for role	When required	Settings" Required privileges in vCenter GUI "Virtual machine".Change Configuration"."Upgrade virtual machine compatibility" "Virtual machine".Interaction."Guest operating system management by VIX API" "Virtual machine".Interaction."Power off" "Virtual machine".Interaction."Power on" "Virtual machine".Interaction.Reset "Virtual machine"."Edit Inventory"."Create new" "Virtual machine"."Edit Inventory"."Create from existing" "Virtual machine"."Edit Inventory"."Remove" "Virtual machine".Provisioning."Clone virtual machine" "Virtual machine".Provisioning."Mark as template" "Virtual machine".Provisioning."Deploy template"
vSphere vCenter data center	The installation program creates the virtual machine folder.	"vSphere Tagging". "Assign or Unassign vSphere Tag on Object" Resource."Assign virtual machine to resource pool" VApp.Import "Virtual machine". "Change Configuration". "Add existing disk" "Virtual machine". "Change Configuration". "Add new disk" "Virtual machine". "Change Configuration". "Add or remove device" "Virtual machine". "Change Configuration". "Advanced configuration" "Virtual machine". "Change Configuration". "Set annotation"

vSphere object for role	When required	Required privileges in vCenter GUI
		<p>"Virtual machine"."Change Configuration"."Change CPU count"</p> <p>"Virtual machine"."Change Configuration"."Extend virtual disk"</p> <p>"Virtual machine"."Change Configuration"."Acquire disk lease"</p> <p>"Virtual machine"."Change Configuration"."Modify device settings"</p> <p>"Virtual machine"."Change Configuration"."Change Memory"</p> <p>"Virtual machine"."Change Configuration"."Remove disk"</p> <p>"Virtual machine"."Change Configuration".Rename</p> <p>"Virtual machine"."Change Configuration"."Reset guest information"</p> <p>"Virtual machine"."Change Configuration"."Change resource"</p> <p>"Virtual machine"."Change Configuration"."Change Settings"</p> <p>"Virtual machine"."Change Configuration"."Upgrade virtual machine compatibility"</p> <p>"Virtual machine".Interaction."Guest operating system management by VIX API"</p> <p>"Virtual machine".Interaction."Power off"</p> <p>"Virtual machine".Interaction."Power on"</p> <p>"Virtual machine".Interaction.Reset</p> <p>"Virtual machine"."Edit Inventory"."Create new"</p> <p>"Virtual machine"."Edit Inventory"."Create from existing"</p> <p>"Virtual machine"."Edit Inventory"."Remove"</p> <p>"Virtual machine".Provisioning."Clone virtual machine"</p>

vSphere object for role	When required	"Virtual machine".Provisioning."Deploy template" "Virtual machine".Provisioning."Mark as template" Folder."Create folder" Folder."Delete folder"
-------------------------	---------------	---

Additionally, the user requires some **ReadOnly** permissions, and some of the roles require permission to propagate the permissions to child objects. These settings vary depending on whether or not you install the cluster into an existing folder.

Example 2.3. Required permissions and propagation settings

vSphere object	When required	Propagate to children	Permissions required
vSphere vCenter	Always	False	Listed required privileges
vSphere vCenter data center	Existing folder	False	ReadOnly permission
	Installation program creates the folder	True	Listed required privileges
vSphere vCenter Cluster	Existing resource pool	False	ReadOnly permission
	VMs in cluster root	True	Listed required privileges
vSphere vCenter Datastore	Always	False	Listed required privileges
vSphere Switch	Always	False	ReadOnly permission
vSphere Port Group	Always	False	Listed required privileges
vSphere vCenter Virtual Machine Folder	Existing folder	True	Listed required privileges
vSphere vCenter Resource Pool	Existing resource pool	True	Listed required privileges

For more information about creating an account with only the required privileges, see [vSphere Permissions and User Management Tasks](#) in the vSphere documentation.

2.1.4.2. Minimum required vCenter account privileges

After you create a custom role and assign privileges to the role, you can create permissions by selecting specific vSphere objects. You can then assign the custom role to a user or group for each object.

Before you create permissions or request for the creation of permissions for a vSphere object, decide what minimum permissions apply to the vSphere object. By doing this task, you can ensure a basic interaction exists between a vSphere object and OpenShift Container Platform architecture.



IMPORTANT

If you create a custom role and you do not assign privileges to it, the vSphere Server by default assigns a **Read Only** role to the custom role. Note that for the cloud provider API, the custom role only needs to inherit the privileges of the **Read Only** role.

Consider creating a custom role when an account with global administrative privileges does not meet your needs.



IMPORTANT

Red Hat does not support configuring an account without including the required privileges. Installing an OpenShift Container Platform cluster in a vCenter is tested against a full list of privileges as described in the "Required vCenter account privileges" section. By adhering to the full list of privileges, you can reduce the possibility of unexpected behaviors that might occur when creating a custom role with a restricted set of privileges.

The following tables specify how the required vCenter account privileges provided earlier in this document are relevant to different aspects of OpenShift Container Platform architecture.

Example 2.4. Minimum permissions on installer-provisioned infrastructure

vSphere object for role	When required	Required privileges

vSphere object for role	When required	Required privileges
vSphere vCenter	Always	Cns.Searchable InventoryService.Tagging.AttachTag InventoryService.Tagging.CreateCategory InventoryService.Tagging.CreateTag InventoryService.Tagging.DeleteCategory InventoryService.Tagging.DeleteTag InventoryService.Tagging.EditCategory InventoryService.Tagging.EditTag Sessions.ValidateSession StorageProfile.Update StorageProfile.View
vSphere vCenter Cluster	If you intend to create VMs in the cluster root	Host.Config.Storage Resource.AssignVMtoPool VApp.AssignResourcePool VApp.Import VirtualMachine.Config .AddNewDisk
vSphere vCenter Resource Pool	If you included an existing resource pool in the <code>install-config.yaml</code> file	Host.Config.Storage Resource.AssignVMtoPool VApp.AssignResourcePool VApp.Import`minimu m
vSphere Datastore	If you referenced a datastore in the <code>install-config.yaml</code> file	Datastore.Browse Datastore.FileManagement InventoryService.Tagging.ObjectAttachable
vSphere Port Group	Always	Network.Assign
Virtual Machine Folder	Always	InventoryService.Tag

vSphere object for role	When required	Required privileges
		<i>VirtualMachine</i> .Config <i>.AddExistingDisk</i> <i>.AddNewDisk</i> <i>.AddRemoveDevice</i> <i>.AdvancedConfig</i> <i>.Annotation</i> <i>.CPUCount</i> <i>.DiskExtend</i> <i>.DiskLease</i> <i>.EditDevice</i> <i>.Memory</i> <i>.RemoveDisk</i> <i>.Rename</i> <i>.ResetGuestInfo</i> <i>.Resource</i> <i>.Settings</i> <i>.UpgradeVirtualHardware</i> <i>VirtualMachine</i> .Interact <i>t.GuestControl</i> <i>VirtualMachine</i> .Interact <i>t.PowerOff</i> <i>VirtualMachine</i> .Interact <i>t.PowerOn</i> <i>VirtualMachine</i> .Interact <i>t.Reset</i> <i>VirtualMachine</i> .Invent <i>ory.Create</i> <i>VirtualMachine</i> .Invent <i>ory.CreateFromExisti ng</i> <i>VirtualMachine</i> .Invent <i>ory.Delete</i> <i>VirtualMachine</i> .Provisi

vSphere object for role	When required	Required privileges
		<i>VirtualMachine.Clone</i> <i>VirtualMachine.MarkAsTemplate</i> <i>VirtualMachine.Provision</i> <i>VirtualMachine.DeployTemplate</i>
vSphere vCenter data center		
		<i>Folder.Create</i> <i>Folder.Delete</i> <i>InventoryService.Tagging</i> <i>ObjectAttachable</i> <i>Resource.AssignVMO</i> <i>oPool</i>
		<i>VApp.Import</i> <i>VirtualMachine.Config</i> <i>.AddExistingDisk</i>
		<i>VirtualMachine.Config</i> <i>.AddNewDisk</i>
		<i>VirtualMachine.Config</i> <i>.AddRemoveDevice</i>
		<i>VirtualMachine.Config</i> <i>.AdvancedConfig</i>
		<i>VirtualMachine.Config</i> <i>.Annotation</i>
		<i>VirtualMachine.Config</i> <i>.CPUCount</i>
		<i>VirtualMachine.Config</i> <i>.DiskExtend</i>
		<i>VirtualMachine.Config</i> <i>.DiskLease</i>
		<i>VirtualMachine.Config</i> <i>.EditDevice</i>
		<i>VirtualMachine.Config</i> <i>.Memory</i>
		<i>VirtualMachine.Config</i> <i>.RemoveDisk</i>
		<i>VirtualMachine.Config</i> <i>.Rename</i>
		<i>VirtualMachine.Config</i> <i>.ResetGuestInfo</i>
		<i>VirtualMachine.Config</i> <i>.Resource</i>
		<i>VirtualMachine.Config</i> <i>.Settings</i>
		<i>VirtualMachine.Config</i> <i>.UpgradeVirtualHardware</i>
		<i>VirtualMachine.Interact</i> <i>.GuestControl</i>
		<i>VirtualMachine.Interact</i> <i>.PowerOff</i>

vSphere object for role	When required	Required privileges
		VirtualMachine.Interaction.PowerOn VirtualMachine.Interaction.Reset VirtualMachine.Inventory.Create VirtualMachine.Inventory.CreateFromExisting VirtualMachine.Inventory.Delete VirtualMachine.Provisioning.Clone VirtualMachine.Provisioning.DeployTemplate VirtualMachine.Provisioning.MarkAsTemplate

Example 2.5. Minimum permissions for postinstallation management of components

vSphere object for role	When required	Required privileges
vSphere vCenter	Always	Cns.Searchable InventoryService.Tagging.AttachTag InventoryService.Tagging.CreateCategory InventoryService.Tagging.CreateTag InventoryService.Tagging.DeleteCategory InventoryService.Tagging.DeleteTag InventoryService.Tagging.EditCategory InventoryService.Tagging.EditTag Sessions.ValidateSession StorageProfile.Update StorageProfile.View
vSphere vCenter Cluster	If you intend to create VMs in the cluster root	Host.Config.Storage Resource.AssignVMPool

vSphere object for role	When required	Required privileges
vSphere vCenter Resource Pool	If you included an existing resource pool in the <code>install-config.yaml</code> file	Host.Config.Storage
vSphere Datastore	Always	Datastore.AllocateSpace Datastore.Browse Datastore.FileManagement InventoryService.Tagging.ObjectAttachable
vSphere Port Group	Always	Network.Assign
Virtual Machine Folder	Always	VirtualMachine.Config.AddExistingDisk VirtualMachine.Config.AddRemoveDevice VirtualMachine.Config.AdvancedConfig VirtualMachine.Config.Annotation VirtualMachine.Config.CPUCount VirtualMachine.Config.DiskExtend VirtualMachine.Config.Memory VirtualMachine.Config.Settings VirtualMachine.Interact.PowerOff VirtualMachine.Interact.PowerOn VirtualMachine.Inventory.CreateFromExisting VirtualMachine.Inventory.Delete VirtualMachine.Provisioning.Clone VirtualMachine.Provisioning.DeployTemplate

vSphere object for role	When required	Required privileges
vSphere vCenter data center	If the virtual machine folder does not already exist, the installation program creates the virtual machine folder.	Resource.AssignVMT oPool VirtualMachine.Config .AddExistingDisk VirtualMachine.Config .AddRemoveDevice VirtualMachine.Interac t.PowerOff VirtualMachine.Interac t.PowerOn VirtualMachine.Provisi oning.DeployTemplat e

Example 2.6. Minimum permissions for the storage components

vSphere object for role	When required	Required privileges
vSphere vCenter	Always	Cns.Searchable InventoryService.Tag ging.CreateCategory InventoryService.Tag ging.CreateTag InventoryService.Tag ging.EditCategory InventoryService.Tag ging.EditTag StorageProfile.Update StorageProfile.View
vSphere vCenter Cluster	If you intend to create VMs in the cluster root	Host.Config.Storage
vSphere vCenter Resource Pool	If you included an existing resource pool in the install-config.yaml file	Host.Config.Storage
vSphere Datastore	Always	Datastore.Browse Datastore.FileManage ment InventoryService.Tag ging.ObjectAttachable
vSphere Port Group	Always	Read Only

vSphere object for role	When required	Required privileges
Virtual Machine Folder	Always	VirtualMachine.Config.AddExistingDisk VirtualMachine.Config.AddRemoveDevice
vSphere vCenter data center	If the virtual machine folder does not already exist, the installation program creates the virtual machine folder.	VirtualMachine.Config.AddExistingDisk VirtualMachine.Config.AddRemoveDevice

Example 2.7. Minimum permissions for the Machine API

vSphere object for role	When required	Required privileges
vSphere vCenter	Always	InventoryService.Tagging.AttachTag InventoryService.Tagging.CreateCategory InventoryService.Tagging.CreateTag InventoryService.Tagging.DeleteCategory InventoryService.Tagging.DeleteTag InventoryService.Tagging.EditCategory InventoryService.Tagging.EditTag Sessions.ValidateSession StorageProfile.Update StorageProfile.View
vSphere vCenter Cluster	If you intend to create VMs in the cluster root	Resource.AssignVMOPool
vSphere vCenter Resource Pool	If you included an existing resource pool in the install-config.yaml file	Read Only
vSphere Datastore	Always	Datastore.AllocateSpace Datastore.Browse
vSphere Port Group	Always	Network.Assign

vSphere object for role	When required	Required privileges
Virtual Machine Folder	Always	VirtualMachine.Config .AddRemoveDevice VirtualMachine.Config .AdvancedConfig VirtualMachine.Config .Annotation VirtualMachine.Config .CPUCount VirtualMachine.Config .DiskExtend VirtualMachine.Config .Memory VirtualMachine.Config .Settings VirtualMachine.Interact .PowerOff VirtualMachine.Interact .PowerOn VirtualMachine.Inventory.CreateFromExisting VirtualMachine.Inventory.Delete VirtualMachine.Provisioning.Clone VirtualMachine.Provisioning.DeployTemplate
vSphere vCenter data center	If the virtual machine folder does not already exist, the installation program creates the virtual machine folder.	Resource.AssignVMT oPool VirtualMachine.Interact .PowerOff VirtualMachine.Interact .PowerOn VirtualMachine.Provisioning.DeployTemplate

2.1.4.3. Using OpenShift Container Platform with vMotion

If you intend on using vMotion in your vSphere environment, consider the following before installing an OpenShift Container Platform cluster.

- Using Storage vMotion can cause issues and is not supported.

- Using VMware compute vMotion to migrate the workloads for both OpenShift Container Platform compute machines and control plane machines is generally supported, where *generally* implies that you meet all VMware best practices for vMotion. To help ensure the uptime of your compute and control plane nodes, ensure that you follow the VMware best practices for vMotion, and use VMware anti-affinity rules to improve the availability of OpenShift Container Platform during maintenance or hardware issues.

For more information about vMotion and anti-affinity rules, see the VMware vSphere documentation for [vMotion networking requirements](#) and [VM anti-affinity rules](#).

- If you are using VMware vSphere volumes in your pods, migrating a VM across datastores, either manually or through Storage vMotion, causes invalid references within OpenShift Container Platform persistent volume (PV) objects that can result in data loss.
- OpenShift Container Platform does not support selective migration of virtual machine disks (VMDKs) across datastores, using datastore clusters for VM provisioning or for dynamic or static provisioning of PVs, or using a datastore that is part of a datastore cluster for dynamic or static provisioning of PVs.



IMPORTANT

You can specify the path of any datastore that exists in a datastore cluster. By default, Storage Distributed Resource Scheduler (SDRS), which uses Storage vMotion, is automatically enabled for a datastore cluster. Red Hat does not support Storage vMotion, so you must disable SDRS to avoid data loss issues for your OpenShift Container Platform cluster.

If you must specify VMs across many datastores, use a **datastore** object to specify a failure domain in your cluster's **install-config.yaml** configuration file. For more information, see "VMware vSphere region and zone enablement".

2.1.4.4. Cluster resources

When you deploy an OpenShift Container Platform cluster that uses installer-provisioned infrastructure, the installation program must be able to create several resources in your vCenter instance.

A standard OpenShift Container Platform installation creates the following vCenter resources:

- 1 Folder
- 1 Tag category
- 1 Tag
- Virtual machines:
 - 1 template
 - 1 temporary bootstrap node
 - 3 control plane nodes
 - 3 compute machines

Although these resources use 856 GB of storage, the bootstrap node gets deleted during the cluster installation process. At a minimum, a standard cluster requires 800 GB of storage.

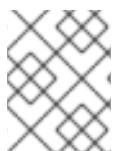
If you deploy more compute machines, the OpenShift Container Platform cluster will use more storage.

2.1.4.5. Cluster limits

Available resources vary between clusters. A limit exists for the number of possible clusters within vCenter, primarily by available storage space and any limitations on the number of required resources. Be sure to consider both limitations to the vCenter resources that the cluster creates and the resources that you require to deploy a cluster, such as IP addresses and networks.

2.1.4.6. Networking requirements

You can use Dynamic Host Configuration Protocol (DHCP) for the network and configure the DHCP server to set persistent IP addresses to machines in your cluster. In the DHCP lease, you must configure the DHCP to use the default gateway.



NOTE

You do not need to use the DHCP for the network if you want to provision nodes with static IP addresses.

If you are installing to a restricted environment, the VM in your restricted network must have access to vCenter so that it can provision and manage nodes, persistent volume claims (PVCs), and other resources.



NOTE

Ensure that each OpenShift Container Platform node in the cluster has access to a Network Time Protocol (NTP) server that is discoverable by DHCP. Installation is possible without an NTP server. However, asynchronous server clocks can cause errors, which the NTP server prevents.

Additionally, you must create the following networking resources before you install the OpenShift Container Platform cluster:

2.1.4.6.1. Required IP addresses

For a network that uses DHCP, an installer-provisioned vSphere installation requires two static IP addresses:

- The **API** address for accessing the cluster API.
- The **Ingress** address for cluster ingress traffic.

You must give these IP addresses to the installation program when you install the OpenShift Container Platform cluster.

2.1.4.6.2. DNS records

You must create DNS records for two static IP addresses in the appropriate DNS server for the vCenter instance that hosts your OpenShift Container Platform cluster. In each record, **<cluster_name>** is the cluster name and **<base_domain>** is the cluster base domain that you specify when you install the cluster. A complete DNS record takes the form: **<component>.<cluster_name>.<base_domain>..**

Table 2.6. Required DNS records

Component	Record	Description
API VIP	<code>api.<cluster_name>.<base_domain>.</code>	This DNS A/AAAA or CNAME (Canonical Name) record must point to the load balancer for the control plane machines. This record must be resolvable by both clients external to the cluster and from all the nodes within the cluster.
Ingress VIP	<code>*.apps.<cluster_name>.<base_domain>.</code>	A wildcard DNS A/AAAA or CNAME record that points to the load balancer that targets the machines that run the Ingress router pods, which are the worker nodes by default. This record must be resolvable by both clients external to the cluster and from all the nodes within the cluster.

2.1.4.6.3. Static IP addresses for vSphere nodes

You can provision bootstrap, control plane, and compute nodes to be configured with static IP addresses in environments where Dynamic Host Configuration Protocol (DHCP) does not exist. To configure this environment, you must provide values to the `platform.vsphere.hosts.role` parameter in the `install-config.yaml` file.

By default, the installation program is configured to use the DHCP for the network, but this network has limited configurable capabilities.

After you define one or more machine pools in your `install-config.yaml` file, you can define network definitions for nodes on your network. Ensure that the number of network definitions matches the number of machine pools that you configured for your cluster.

Example network configuration that specifies different roles

```
# ...
platform:
  vsphere:
    hosts:
      - role: bootstrap 1
        networkDevice:
          ipAddrs:
            - 192.168.204.10/24 2
          gateway: 192.168.204.1 3
          nameservers:
            - 192.168.204.1
      - role: control-plane
        networkDevice:
          ipAddrs:
```

```

- 192.168.204.11/24
  gateway: 192.168.204.1
  nameservers:
    - 192.168.204.1
- role: control-plane
  networkDevice:
    ipAddrs:
      - 192.168.204.12/24
    gateway: 192.168.204.1
    nameservers:
      - 192.168.204.1
- role: control-plane
  networkDevice:
    ipAddrs:
      - 192.168.204.13/24
    gateway: 192.168.204.1
    nameservers:
      - 192.168.204.1
- role: compute
  networkDevice:
    ipAddrs:
      - 192.168.204.14/24
    gateway: 192.168.204.1
    nameservers:
      - 192.168.204.1
# ...

```

- 1 Valid network definition values include **bootstrap**, **control-plane**, and **compute**. You must list at least one **bootstrap** network definition in your **install-config.yaml** configuration file.
- 2 Lists IPv4, IPv6, or both IP addresses that the installation program passes to the network interface. The machine API controller assigns all configured IP addresses to the default network interface.
- 3 The default gateway for the network interface.
- 4 Lists up to 3 DNS nameservers.

After you deployed your cluster to run nodes with static IP addresses, you can scale a machine to use one of these static IP addresses. Additionally, you can use a machine set to configure a machine to use one of the configured static IP addresses.

Additional resources

- Scaling machines to use static IP addresses
- Using a machine set to scale machines with configured static IP addresses

2.2. PREPARING TO INSTALL A CLUSTER USING INSTALLER-PROVISIONED INFRASTRUCTURE

You prepare to install an OpenShift Container Platform cluster on vSphere by completing the following steps:

- Downloading the installation program.

**NOTE**

If you are installing in a disconnected environment, you extract the installation program from the mirrored content. For more information, see [Mirroring images for a disconnected installation](#).

- Installing the OpenShift CLI (**oc**).

**NOTE**

If you are installing in a disconnected environment, install **oc** to the mirror host.

- Generating an SSH key pair. You can use this key pair to authenticate into the OpenShift Container Platform cluster's nodes after it is deployed.
- Adding your vCenter's trusted root CA certificates to your system trust.

2.2.1. Obtaining the installation program

Before you install OpenShift Container Platform, download the installation file on the host you are using for installation.

Prerequisites

- You have a machine that runs Linux, for example Red Hat Enterprise Linux 8, with 500 MB of local disk space.

**IMPORTANT**

If you attempt to run the installation program on macOS, a known issue related to the **golang** compiler causes the installation of the OpenShift Container Platform cluster to fail. For more information about this issue, see the section named "Known Issues" in the *OpenShift Container Platform 4.19 release notes* document.

Procedure

1. Go to the [Cluster Type](#) page on the Red Hat Hybrid Cloud Console. If you have a Red Hat account, log in with your credentials. If you do not, create an account.

TIP

You can also [download the binaries for a specific OpenShift Container Platform release](#).

2. Select your infrastructure provider from the **Run it yourself** section of the page.
3. Select your host operating system and architecture from the dropdown menus under **OpenShift Installer** and click **Download Installer**.
4. Place the downloaded file in the directory where you want to 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 the installation program and the files that the installation program creates after you finish installing the cluster. Both of the files are required to delete the cluster.
- Deleting the files created by the installation program does not remove your cluster, even if the cluster failed during installation. To remove your cluster, complete the OpenShift Container Platform uninstallation procedures for your specific cloud provider.

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

```
$ tar -xvf openshift-install-linux.tar.gz
```

6. Download your installation [pull secret from Red Hat OpenShift Cluster Manager](#). 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.

TIP

Alternatively, you can retrieve the installation program from the [Red Hat Customer Portal](#), where you can specify a version of the installation program to download. However, you must have an active subscription to access this page.

2.2.2. Installing the OpenShift CLI on Linux

To manage your cluster and deploy applications from the command line, install the OpenShift CLI (**oc**) binary on Linux.



IMPORTANT

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

Download and install the new version of **oc**.

Procedure

1. Navigate to the [Download OpenShift Container Platform](#) page on the Red Hat Customer Portal.
2. Select the architecture from the **Product Variant** list.
3. Select the appropriate version from the **Version** list.
4. Click **Download Now** next to the **OpenShift v4.19 Linux Clients** entry and save the file.
5. Unpack the archive:

```
$ tar xvf <file>
```

6. Place the **oc** binary in a directory that is on your **PATH**.

To check your **PATH**, execute the following command:

```
$ echo $PATH
```

Verification

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

```
$ oc <command>
```

2.2.3. Installing the OpenShift CLI on Windows

To manage your cluster and deploy applications from the command line, install OpenShift CLI (**oc**) binary on Windows.



IMPORTANT

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

Download and install the new version of **oc**.

Procedure

1. Navigate to the [Download OpenShift Container Platform](#) page on the Red Hat Customer Portal.
2. Select the appropriate version from the **Version** list.
3. Click **Download Now** next to the **OpenShift v4.19 Windows Client** entry and save the file.
4. Extract the archive with a ZIP program.
5. Move the **oc** binary to a directory that is on your **PATH** variable.

To check your **PATH** variable, open the command prompt and execute the following command:

```
C:> path
```

Verification

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

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

2.2.4. Installing the OpenShift CLI on macOS

To manage your cluster and deploy applications from the command line, install the OpenShift CLI (**oc**) binary on macOS.



IMPORTANT

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

Download and install the new version of **oc**.

Procedure

1. Navigate to the [Download OpenShift Container Platform](#) page on the Red Hat Customer Portal.
2. Select the architecture from the **Product Variant** list.
3. Select the appropriate version from the **Version** list.
4. Click **Download Now** next to the **OpenShift v4.19 macOS Clients** entry and save the file.



NOTE

For macOS arm64, choose the **OpenShift v4.19 macOS arm64 Client** entry.

5. Unpack and unzip the archive.
6. Move the **oc** binary to a directory on your **PATH** variable.

To check your **PATH** variable, open a terminal and execute the following command:

```
$ echo $PATH
```

Verification

- Verify your installation by using an **oc** command:

```
$ oc <command>
```

2.2.5. Generating a key pair for cluster node SSH access

To enable secure, passwordless SSH access to your cluster nodes, provide an SSH public key during the OpenShift Container Platform installation. This ensures that the installation program automatically configures the Red Hat Enterprise Linux CoreOS (RHCOS) nodes for remote authentication through the **core** user.

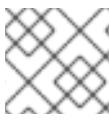
The SSH public key gets added to the `~/.ssh/authorized_keys` list for the **core** user on each node. After the key is passed to the Red Hat Enterprise Linux CoreOS (RHCOS) nodes through their Ignition config files, you can use the key pair to SSH in to the RHCOS nodes as the user **core**. To access the nodes through SSH, the private key identity must be managed by SSH for your local user.

If you want to SSH in to your cluster nodes to perform installation debugging or disaster recovery, you must provide the SSH public key during the installation process. The `./openshift-install gather` command also requires the SSH public key to be in place on the cluster nodes.



IMPORTANT

Do not skip this procedure in production environments, where disaster recovery and debugging is required.



NOTE

You must use a local key, not one that you configured with platform-specific approaches.

Procedure

1. If you do not have an existing SSH key pair on your local machine to use for authentication onto your cluster nodes, create one. For example, on a computer that uses a Linux operating system, run the following command:

```
$ ssh-keygen -t ed25519 -N "" -f <path>/<file_name>
```

Specifies the path and file name, such as `~/.ssh/id_ed25519`, of the new SSH key. If you have an existing key pair, ensure your public key is in the your `~/.ssh` directory.



NOTE

If you plan to install an OpenShift Container Platform cluster that uses 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, do not create a key that uses the **ed25519** algorithm. Instead, create a key that uses the **rsa** or **ecdsa** algorithm.

2. View the public SSH key:

```
$ cat <path>/<file_name>.pub
```

For example, run the following to view the `~/.ssh/id_ed25519.pub` public key:

```
$ cat ~/.ssh/id_ed25519.pub
```

3. Add the SSH private key identity to the SSH agent for your local user, if it has not already been added. SSH agent management of the key is required for password-less SSH authentication onto your cluster nodes, or if you want to use the `./openshift-install gather` command.



NOTE

On some distributions, default SSH private key identities such as `~/.ssh/id_rsa` and `~/.ssh/id_dsa` are managed automatically.

- a. If the **ssh-agent** process is not already running for your local user, start it as a background task:

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

Example output



Agent pid 31874



NOTE

If your cluster is in FIPS mode, only use FIPS-compliant algorithms to generate the SSH key. The key must be either RSA or ECDSA.

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

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

Specifies the path and file name for your SSH private key, such as `~/.ssh/id_ed25519`

Example output

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

Next steps

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

2.2.6. Adding vCenter root CA certificates to your system trust

Because the installation program requires access to your vCenter’s API, you must add your vCenter’s trusted root CA certificates to your system trust before you install an OpenShift Container Platform cluster.

Procedure

- From the vCenter home page, download the vCenter’s root CA certificates. Click **Download trusted root CA certificates** in the vSphere Web Services SDK section. The `<vCenter>/certs/download.zip` file downloads.
- Extract the compressed file that contains the vCenter root CA certificates. The contents of the compressed file resemble the following file structure:

```

certs
└── lin
    ├── 108f4d17.0
    ├── 108f4d17.r1
    ├── 7e757f6a.0
    ├── 8e4f8471.0
    └── 8e4f8471.r0
└── mac
    ├── 108f4d17.0
    ├── 108f4d17.r1
    ├── 7e757f6a.0
    ├── 8e4f8471.0
    └── 8e4f8471.r0
└── win
    ├── 108f4d17.0.crt
    └── 108f4d17.r1.crl

```

```

    └── 7e757f6a.0.crt
    └── 8e4f8471.0.crt
    └── 8e4f8471.r0.crl

3 directories, 15 files

```

3. Add the files for your operating system to the system trust. For example, on a Fedora operating system, run the following command:

```
# cp certs/lin/* /etc/pki/ca-trust/source/anchors
```

4. Update your system trust. For example, on a Fedora operating system, run the following command:

```
# update-ca-trust extract
```

2.3. INSTALLING A CLUSTER ON VSPHERE

In OpenShift Container Platform version 4.19, you can install a cluster on your VMware vSphere instance by using installer-provisioned infrastructure.

2.3.1. Prerequisites

- You have completed the tasks in [Preparing to install a cluster using installer-provisioned infrastructure](#).
- You reviewed your VMware platform licenses. Red Hat does not place any restrictions on your VMware licenses, but some VMware infrastructure components require licensing.
- You reviewed details about the [OpenShift Container Platform installation and update](#) processes.
- You read the documentation on [selecting a cluster installation method and preparing it for users](#).
- You provisioned [persistent storage](#) for your cluster. To deploy a private image registry, your storage must provide **ReadWriteMany** access modes.
- The OpenShift Container Platform installer requires access to port 443 on the vCenter and ESXi hosts. You verified that port 443 is accessible.
- If you use a firewall, you confirmed with the administrator that port 443 is accessible. Control plane nodes must be able to reach vCenter and ESXi hosts on port 443 for the installation to succeed.
- If you use a firewall, you [configured it to allow the sites](#) that your cluster requires access to.



NOTE

Be sure to also review this site list if you are configuring a proxy.

2.3.2. Internet access for OpenShift Container Platform

In OpenShift Container Platform 4.19, you require access to the internet to install your cluster.

You must have internet access to perform the following actions:

- Access [OpenShift Cluster Manager](#) to download the installation program and perform subscription management. If the cluster has internet access and you do not disable Telemetry, that service automatically entitles your cluster.
- 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 required content and use it to populate a mirror registry with the installation packages. 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.

2.3.3. Deploying 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

- You have the OpenShift Container Platform installation program and the pull secret for your cluster.
- You have verified that the cloud provider account on your host has the correct permissions to deploy the cluster. An account with incorrect permissions causes the installation process to fail with an error message that displays the missing permissions.
- Optional: Before you create the cluster, you configured an external load balancer in place of the default load balancer.



IMPORTANT

You do not need to specify API and Ingress static addresses for your installation program. If you choose this configuration, you must take additional actions to define network targets that accept an IP address from each referenced vSphere subnet. See the section "Configuring a user-managed load balancer".

Procedure

1. In the directory that contains the installation program, initialize the cluster deployment by running the following command:

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

- 1 For **<installation_directory>**, specify the directory name to store the files that the installation program creates.
- 2 To view different installation details, specify **warn**, **debug**, or **error** instead of **info**.

When specifying the directory:

- Verify that the directory has the **execute** permission. This permission is required to run Terraform binaries under the installation directory.
- Use an empty directory. Some installation assets, such as bootstrap X.509 certificates, have short expiration intervals, therefore 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.

2. Provide values at the prompts:

- a. 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, specify an SSH key that your **ssh-agent** process uses.

- b. Select **vsphere** as the platform to target.
- c. Specify the name of your vCenter instance.
- d. Specify the user name and password for the vCenter account that has the required permissions to create the cluster.

The installation program connects to your vCenter instance.



IMPORTANT

Some VMware vCenter Single Sign-On (SSO) environments with Active Directory (AD) integration might primarily require you to use the traditional login method, which requires the **<domain>** construct.

To ensure that vCenter account permission checks complete properly, consider using the User Principal Name (UPN) login method, such as **<username>@<fully_qualified_domainname>**.

- e. Select the data center in your vCenter instance to connect to.
- f. Select the default vCenter datastore to use.



NOTE

Datastore and cluster names cannot exceed 60 characters; therefore, ensure the combined string length does not exceed the 60 character limit.

- g. Select the vCenter cluster to install the OpenShift Container Platform cluster in. The installation program uses the root resource pool of the vSphere cluster as the default resource pool.
- h. Select the network in the vCenter instance that contains the virtual IP addresses and DNS records that you configured.
- i. Enter the virtual IP address that you configured for control plane API access.
- j. Enter the virtual IP address that you configured for cluster ingress.
- k. Enter the base domain. This base domain must be the same one that you used in the DNS records that you configured.
- l. Enter a descriptive name for your cluster. The cluster name must be the same one that you used in the DNS records that you configured.



NOTE

Datastore and cluster names cannot exceed 60 characters; therefore, ensure the combined string length does not exceed the 60 character limit.

- m. Paste the [pull secret from Red Hat OpenShift Cluster Manager](#).

Verification

When the cluster deployment completes successfully:

- The terminal displays directions for accessing your cluster, including a link to the web console and credentials for the **kubeadmin** user.
- Credential information also outputs to **<installation_directory>/openshift_install.log**.



IMPORTANT

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

Example output

```
...
INFO Install complete!
INFO To access the cluster as the system:admin user when using 'oc', run 'export
KUBECONFIG=/home/myuser/install_dir/auth/kubeconfig'
INFO Access the OpenShift web-console here: https://console-openshift-
console.apps.mycluster.example.com
INFO Login to the console with user: "kubeadmin", and password: "password"
INFO Time elapsed: 36m22s
```



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.
- It is recommended that you use 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.

2.3.4. Logging in to the cluster by using the CLI

To log in to your cluster as the default system user, export the **kubeconfig** file. This configuration enables the CLI to authenticate and connect to the specific API server created during OpenShift Container Platform installation.

The **kubeconfig** file is specific to a cluster and is created during OpenShift Container Platform installation.

Prerequisites

- You deployed an OpenShift Container Platform cluster.
- You installed the OpenShift CLI (**oc**).

Procedure

1. Export the **kubeadmin** credentials by running the following command:

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

where:

<installation_directory>

Specifies the path to the directory that stores the installation files.

2. Verify you can run **oc** commands successfully using the exported configuration by running the following command:

```
$ oc whoami
```

Example output

```
system:admin
```

2.3.5. Creating registry storage

After you install the cluster, you must create storage for the registry Operator.

2.3.5.1. Image registry removed during installation

On platforms that do not provide shareable object storage, the OpenShift Image Registry Operator bootstraps itself as **Removed**. This allows **openshift-installer** to complete installations on these platform types.

After installation, you must edit the Image Registry Operator configuration to switch the **managementState** from **Removed** to **Managed**. When this has completed, you must configure storage.

2.3.5.2. Image registry storage configuration

The Image Registry Operator is not initially available for platforms that do not provide default storage. After installation, you must configure your registry to use storage so that the Registry Operator is made available.

Configure a persistent volume, which is required for production clusters. Where applicable, you can configure an empty directory as the storage location for non-production clusters.

You can also allow the image registry to use block storage types by using the **Recreate** rollout strategy during upgrades.

2.3.5.2.1. Configuring registry storage for VMware vSphere

As a cluster administrator, following installation you must configure your registry to use storage.

Prerequisites

- Cluster administrator permissions.
- A cluster on VMware vSphere.
- Persistent storage provisioned for your cluster, such as Red Hat OpenShift Data Foundation.



IMPORTANT

OpenShift Container Platform supports **ReadWriteOnce** access for image registry storage when you have only one replica. **ReadWriteOnce** access also requires that the registry uses the **Recreate** rollout strategy. To deploy an image registry that supports high availability with two or more replicas, **ReadWriteMany** access is required.

- Must have "100Gi" capacity.



IMPORTANT

Testing shows issues with using the NFS server on RHEL as storage backend for core services. This includes the OpenShift Container Registry and Quay, Prometheus for monitoring storage, and Elasticsearch for logging storage. Therefore, using RHEL NFS to back PVs used by core services is not recommended.

Other NFS implementations on the marketplace might not have these issues. Contact the individual NFS implementation vendor for more information on any testing that was possibly completed against these OpenShift Container Platform core components.

Procedure

1. Change the **spec.storage.pvc** field in the **configs.imageregistry/cluster** resource.



NOTE

When you use shared storage, review your security settings to prevent outside access.

2. Verify that you do not have a registry pod by running the following command:

```
$ oc get pod -n openshift-image-registry -l docker-registry=default
```

Example output

```
No resources found in openshift-image-registry namespace
```



NOTE

If you do have a registry pod in your output, you do not need to continue with this procedure.

3. Check the registry configuration by running the following command:

```
$ oc edit configs.imageregistry.operator.openshift.io
```

Example output

```
storage:  
pvc:  
claim:
```

Leave the **claim** field blank to allow the automatic creation of an **image-registry-storage** persistent volume claim (PVC). The PVC is generated based on the default storage class. However, be aware that the default storage class might provide ReadWriteOnce (RWO) volumes, such as a RADOS Block Device (RBD), which can cause issues when you replicate to more than one replica.

4. Check the **clusteroperator** status by running the following command:

```
$ oc get clusteroperator image-registry
```

Example output

NAME	VERSION	AVAILABLE	PROGRESSING	DEGRADED	SINCE
MESSAGE					
image-registry	4.7	True	False	False	6h50m

2.3.5.2.2. Configuring block registry storage for VMware vSphere

To allow the image registry to use block storage types such as vSphere Virtual Machine Disk (VMDK) during upgrades as a cluster administrator, you can use the **Recreate** rollout strategy.



IMPORTANT

Block storage volumes are supported but not recommended for use with image registry on production clusters. An installation where the registry is configured on block storage is not highly available because the registry cannot have more than one replica.

Procedure

1. Enter the following command to set the image registry storage as a block storage type, patch the registry so that it uses the **Recreate** rollout strategy, and runs with only **1** replica:

```
$ oc patch config.imageregistry.operator.openshift.io/cluster --type=merge -p '{"spec": {"rolloutStrategy": "Recreate", "replicas": 1}}'
```

2. Provision the persistent volume (PV) for the block storage device, and create a persistent volume claim (PVC) for that volume. The requested block volume uses the `ReadWriteOnce` (RWO) access mode.
 - a. Create a **pvc.yaml** file with the following contents to define a VMware vSphere **PersistentVolumeClaim** object:

```
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: image-registry-storage
  namespace: openshift-image-registry
spec:
  accessModes:
  - ReadWriteOnce
  resources:
    requests:
      storage: 100Gi
```

where:

name

Specifies a unique name that represents the **PersistentVolumeClaim** object.

namespace

Specifies the **namespace** for the **PersistentVolumeClaim** object, which is **openshift-image-registry**.

accessModes

Specifies the access mode of the persistent volume claim. With **ReadWriteOnce**, the volume can be mounted with read and write permissions by a single node.

storage

The size of the persistent volume claim.

- b. Enter the following command to create the **PersistentVolumeClaim** object from the file:

```
$ oc create -f pvc.yaml -n openshift-image-registry
```

3. Enter the following command to edit the registry configuration so that it references the correct PVC:

```
$ oc edit config.imageregistry.operator.openshift.io -o yaml
```

Example output

```
storage:  
pvc:  
claim:
```

By creating a custom PVC, you can leave the **claim** field blank for the default automatic creation of an **image-registry-storage** PVC.

For instructions about configuring registry storage so that it references the correct PVC, see [Configuring the registry for vSphere](#).

2.3.6. Telemetry access for OpenShift Container Platform

To provide metrics about cluster health and the success of updates, the Telemetry service requires internet access. When connected, this service runs automatically by default and registers your cluster to [OpenShift Cluster Manager](#).

After you confirm that your [OpenShift Cluster Manager](#) inventory is correct, either maintained automatically by Telemetry or manually by using OpenShift Cluster Manager, use subscription watch to track your OpenShift Container Platform subscriptions at the account or multi-cluster level. For more information about subscription watch, see "Data Gathered and Used by Red Hat's subscription services" in the *Additional resources* section.

Additional resources

- See [About remote health monitoring](#) for more information about the Telemetry service

2.3.7. Next steps

- [Customize your cluster](#).
- If necessary, you can [Remote health reporting](#).
- [Set up your registry and configure registry storage](#).
- Optional: [View the events from the vSphere Problem Detector Operator](#) to determine if the cluster has permission or storage configuration issues.

2.4. INSTALLING A CLUSTER ON VSphere WITH CUSTOMIZATIONS

In OpenShift Container Platform version 4.19, you can install a cluster on your VMware vSphere instance by using installer-provisioned infrastructure. To customize the installation, you modify parameters in the **install-config.yaml** file before you install the cluster.

2.4.1. Prerequisites

- You have completed the tasks in [Preparing to install a cluster using installer-provisioned infrastructure](#).
- You reviewed your VMware platform licenses. Red Hat does not place any restrictions on your VMware licenses, but some VMware infrastructure components require licensing.
- You reviewed details about the [OpenShift Container Platform installation and update](#) processes.
- You read the documentation on [selecting a cluster installation method and preparing it for users](#).
- You provisioned [persistent storage](#) for your cluster. To deploy a private image registry, your storage must provide **ReadWriteMany** access modes.
- The OpenShift Container Platform installer requires access to port 443 on the vCenter and ESXi hosts. You verified that port 443 is accessible.
- If you use a firewall, you confirmed with the administrator that port 443 is accessible. Control plane nodes must be able to reach vCenter and ESXi hosts on port 443 for the installation to succeed.
- If you use a firewall, you [configured it to allow the sites](#) that your cluster requires access to.



NOTE

Be sure to also review this site list if you are configuring a proxy.

2.4.2. Internet access for OpenShift Container Platform

In OpenShift Container Platform 4.19, you require access to the internet to install your cluster.

You must have internet access to perform the following actions:

- Access [OpenShift Cluster Manager](#) to download the installation program and perform subscription management. If the cluster has internet access and you do not disable Telemetry, that service automatically entitles your cluster.
- Access [Quay.io](#) to obtain the packages that are required to install your cluster.
- Obtain the packages that are required to perform cluster updates.

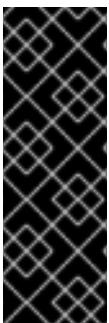


IMPORTANT

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 required content and use it to populate a mirror registry with the installation packages. 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.

2.4.3. VMware vSphere region and zone enablement

You can deploy an OpenShift Container Platform cluster to multiple vSphere data centers. Each data center can run multiple clusters. This configuration reduces the risk of a hardware failure or network outage that can cause your cluster to fail. To enable regions and zones, you must define multiple failure domains for your OpenShift Container Platform cluster.



IMPORTANT

The VMware vSphere region and zone enablement feature requires the vSphere Container Storage Interface (CSI) driver as the default storage driver in the cluster. As a result, the feature is only available on a newly installed cluster.

For a cluster that was upgraded from a previous release, you must enable CSI automatic migration for the cluster. You can then configure multiple regions and zones for the upgraded cluster.

The default installation configuration deploys a cluster to a single vSphere data center. If you want to deploy a cluster to multiple vSphere data centers, you must create an installation configuration file that enables the region and zone feature.

The default **install-config.yaml** file includes **vcenters** and **failureDomains** fields, where you can specify multiple vSphere data centers and clusters for your OpenShift Container Platform cluster. You can leave these fields blank if you want to install an OpenShift Container Platform cluster in a vSphere environment that consists of single data center.

The following list describes terms associated with defining zones and regions for your cluster:

- Failure domain: Establishes the relationships between a region and zone. You define a failure domain by using vCenter objects, such as a **datastore** object. A failure domain defines the vCenter location for OpenShift Container Platform cluster nodes.
- Region: Specifies a vCenter data center. You define a region by using a tag from the **openshift-region** tag category.
- Zone: Specifies a vCenter cluster. You define a zone by using a tag from the **openshift-zone** tag category.



NOTE

If you plan on specifying more than one failure domain in your **install-config.yaml** file, you must create tag categories, zone tags, and region tags in advance of creating the configuration file.

You must create a vCenter tag for each vCenter data center, which represents a region. Additionally, you must create a vCenter tag for each cluster than runs in a data center, which represents a zone. After you create the tags, you must attach each tag to their respective data centers and clusters.

The following table outlines an example of the relationship among regions, zones, and tags for a configuration with multiple vSphere data centers running in a single VMware vCenter.

Data center (region)	Cluster (zone)	Tags
us-east	us-east-1	us-east-1a
		us-east-1b
	us-east-2	us-east-2a
		us-east-2b
us-west	us-west-1	us-west-1a
		us-west-1b
	us-west-2	us-west-2a
		us-west-2b

2.4.4. VMware vSphere host group enablement



IMPORTANT

OpenShift zones support for vSphere host groups is a Technology Preview feature only. Technology Preview features are not supported with Red Hat production service level agreements (SLAs) and might not be functionally complete. Red Hat does not recommend using them in production. These features provide early access to upcoming product features, enabling customers to test functionality and provide feedback during the development process.

For more information about the support scope of Red Hat Technology Preview features, see [Technology Preview Features Support Scope](#).

When deploying an OpenShift Container Platform cluster to VMware vSphere, you can map your vSphere host groups onto OpenShift Container Platform failure domains. This is useful if you are using a stretched cluster configuration, where ESXi hosts are grouped into host groups by physical location.

To enable this feature, you must meet the following requirements:

- You must arrange your ESXi hosts into host groups.
- You must create a vCenter tag in the **openshift-region** tag category for your cluster. After you create the tag, you must attach the tag to the cluster.

- You must create a vCenter tag in the **openshift-zone** tag category for each host group and then attach the correct tag to each ESXi host.
- You must define multiple failure domains for your OpenShift Container Platform cluster in the **install-config.yaml** file.
- You must grant the **Host.Inventory.EditCluster** privilege on the vSphere vCenter cluster object.
- You must include the following parameters in your **install-config.yaml** file to enable this Technology Preview feature:

```
featureSet: TechPreviewNoUpgrade
featureGate:
  - "VSphereHostVMGroupZonal=true"
```



NOTE

For further information on feature gates, see "Enabling features using feature gates".

Review the following key terms, which correspond to parameters in your **install-config.yaml** file that you must configure to enable this feature:

- Failure domain: Specifies the relationships between regions and zones in OpenShift Container Platform, and clusters and host groups in vSphere. You define a failure domain by using vCenter objects, such as a **datastore** object. A failure domain defines the vCenter location for OpenShift Container Platform cluster nodes.
- Region: Specifies a vCenter cluster. You define a region by using a tag from the **openshift-region** tag category.
- Zone: Specifies a vCenter host group. You define a zone by using a tag from the **openshift-zone** tag category.
- Region type: Specifies the **ComputeCluster** region type to enable this feature.
- Zone type: Specifies the **HostGroup** zone type to enable this feature.

Additional resources

- [Additional VMware vSphere configuration parameters](#)
- [Deprecated VMware vSphere configuration parameters](#)
- [vSphere automatic migration](#)
- [VMware vSphere CSI Driver Operator](#)
- [Enabling features using feature gates](#)

2.4.5. Creating the installation configuration file

You can customize the OpenShift Container Platform cluster you install on

VMware vSphere.

Prerequisites

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

Procedure

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

- a. Change to the directory that contains the installation program and run the following command:

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

- **<installation_directory>**: For **<installation_directory>**, specify the directory name to store the files that the installation program creates.
When specifying the directory:
 - Verify that the directory has the **execute** permission. This permission is required to run Terraform binaries under the installation directory.
 - Use an empty directory. Some installation assets, such as bootstrap X.509 certificates, have short expiration intervals, therefore 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, specify an SSH key that your **ssh-agent** process uses.

- ii. Select **vsphere** as the platform to target.
- iii. Specify the name of your vCenter instance.
- iv. Specify the user name and password for the vCenter account that has the required permissions to create the cluster.
The installation program connects to your vCenter instance.
- v. Select the data center in your vCenter instance to connect to.



NOTE

After you create the installation configuration file, you can modify the file to create a multiple vSphere data center environment. This means that you can deploy an OpenShift Container Platform cluster to multiple vSphere data centers. For more information about creating this environment, see the section named *VMware vSphere region and zone enablement*.

- vi. Select the default vCenter datastore to use.



WARNING

You can specify the path of any datastore that exists in a datastore cluster. By default, Storage Distributed Resource Scheduler (SDRS), which uses Storage vMotion, is automatically enabled for a datastore cluster. Red Hat does not support Storage vMotion, so you must disable Storage DRS to avoid data loss issues for your OpenShift Container Platform cluster.

You cannot specify more than one datastore path. If you must specify VMs across multiple datastores, use a **datastore** object to specify a failure domain in your cluster's **install-config.yaml** configuration file. For more information, see "VMware vSphere region and zone enablement".

- vii. Select the vCenter cluster to install the OpenShift Container Platform cluster in. The installation program uses the root resource pool of the vSphere cluster as the default resource pool.
- viii. Select the network in the vCenter instance that contains the virtual IP addresses and DNS records that you configured.
- ix. Enter the virtual IP address that you configured for control plane API access.
- x. Enter the virtual IP address that you configured for cluster ingress.
- xi. Enter the base domain. This base domain must be the same one that you used in the DNS records that you configured.
- xii. Enter a descriptive name for your cluster.
The cluster name you enter must match the cluster name you specified when configuring the DNS records.

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

**NOTE**

If you are installing a three-node cluster, be sure to set the **compute.replicas** parameter to **0**. This ensures that the cluster's control planes are schedulable. For more information, see "Installing a three-node cluster on vSphere".

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.

Additional resources

- [Installation configuration parameters](#)

2.4.5.1. Sample install-config.yaml file for an installer-provisioned VMware vSphere cluster

You can customize the **install-config.yaml** file to specify more details about your OpenShift Container Platform cluster's platform or change the values of the required parameters.

```
apiVersion: v1
baseDomain: example.com 1
compute: 2
- architecture: amd64
  name: <worker_node>
  platform: {}
  replicas: 3
controlPlane: 3
  architecture: amd64
  name: <parent_node>
  platform: {}
  replicas: 3
metadata:
  creationTimestamp: null
  name: test 4
platform:
  vsphere: 5
    apiVIPs:
    - 10.0.0.1
    failureDomains: 6
    - name: <failure_domain_name>
      region: <default_region_name>
      server: <fully_qualified_domain_name>
    topology:
      computeCluster: "<data_center>/host/<cluster>"
      datacenter: <data_center>
      datastore: "<data_center>/datastore/<datastore>" 7
      networks:
      - <VM_Network_name>
      resourcePool: "<data_center>/host/<cluster>/Resources/<resourcePool>" 8
      folder: "<data_center_name>/vm/<folder_name>/<subfolder_name>"
```

```

tagIDs: 9
- <tag_id> 10
zone: <default_zone_name>
ingressVIPs:
- 10.0.0.2
vcenters:
- datacenters:
- <data_center>
password: <password>
port: 443
server: <fully_qualified_domain_name>
user: administrator@vsphere.local
diskType: thin 11
fips: false
pullSecret: '{"auths": ...}'
sshKey: 'ssh-ed25519 AAAA...'
```

- 1 The base domain of the cluster. All DNS records must be sub-domains of this base and include the cluster name.
- 2 3 The **controlPlane** section is a single mapping, but the **compute** section is a sequence of mappings. To meet the requirements of the different data structures, the first line of the **compute** section must begin with a hyphen, -, and the first line of the **controlPlane** section must not. Only one control plane pool is used.
- 4 The cluster name that you specified in your DNS records.
- 5 Optional: Provides additional configuration for the machine pool parameters for the compute and control plane machines.



IMPORTANT

The VIPs, **apiVIP** and **ingressVIP**, must come from the same **networking.machineNetwork** segment. For **apiVIP** and for **ingressVIP**, if the **networking.machineNetwork** is **10.0.0.0/16** then API VIPs and Ingress VIPs must be in one of the **10.0.0.0/16** machine networks.

- 6 Establishes the relationships between a region and zone. You define a failure domain by using vCenter objects, such as a **datastore** object. A failure domain defines the vCenter location for OpenShift Container Platform cluster nodes.
- 7 The path to the vSphere datastore that holds virtual machine files, templates, and ISO images.



IMPORTANT

You can specify the path of any datastore that exists in a datastore cluster. By default, Storage vMotion is automatically enabled for a datastore cluster. Red Hat does not support Storage vMotion, so you must disable Storage vMotion to avoid data loss issues for your OpenShift Container Platform cluster.

If you must specify VMs across multiple datastores, use a **datastore** object to specify a failure domain in your cluster's **install-config.yaml** configuration file. For more information, see "VMware vSphere region and zone enablement".

- 8 Optional: Provides an existing resource pool for machine creation. If you do not specify a value, the installation program uses the root resource pool of the vSphere cluster.
- 9 Optional: Each VM created by OpenShift Container Platform is assigned a unique tag that is specific to the cluster. The assigned tag enables the installation program to identify and remove the associated VMs when a cluster is decommissioned. You can list up to ten additional tag IDs to be attached to the VMs provisioned by the installation program.
- 10 The ID of the tag to be associated by the installation program. For example, **urn:vmomi:InventoryServiceTag:208e713c-cae3-4b7f-918e-4051ca7d1f97:GLOBAL**. For more information about determining the tag ID, see the [vSphere Tags and Attributes documentation](#).
- 11 The vSphere disk provisioning method.



NOTE

In OpenShift Container Platform 4.12 and later, the **apiVIP** and **ingressVIP** configuration settings are deprecated. Instead, use a list format to enter values in the **apiVIPs** and **ingressVIPs** configuration settings.

2.4.5.2. Configuring the cluster-wide proxy during installation

To enable internet access in environments that deny direct connections, configure a cluster-wide proxy in the **install-config.yaml** file. This configuration ensures that the new OpenShift Container Platform cluster routes traffic through the specified HTTP or HTTPS proxy.

Prerequisites

- You have an existing **install-config.yaml** file.
- You have reviewed the sites that your cluster requires access to and determined whether any of them need to bypass the proxy. By default, all cluster egress traffic is proxied, including calls to hosting cloud provider APIs. You added sites to the **Proxy** object's **spec.noProxy** field to bypass the proxy if necessary.



NOTE

The **Proxy** object **status.noProxy** field is populated with the values of the **networking.machineNetwork[].cidr**, **networking.clusterNetwork[].cidr**, and **networking.serviceNetwork[]** fields from your installation configuration.

For installations on Amazon Web Services (AWS), Google Cloud, Microsoft Azure, and Red Hat OpenStack Platform (RHOSP), the **Proxy** object **status.noProxy** field is also populated with the instance metadata endpoint (**169.254.169.254**).

Procedure

1. Edit your **install-config.yaml** file and add the proxy settings. For example:

```
apiVersion: v1
baseDomain: my.domain.com
proxy:
```

```

httpProxy: http://<username>:<pswd>@<ip>:<port>
httpsProxy: https://<username>:<pswd>@<ip>:<port>
noProxy: example.com
additionalTrustBundle: |
  -----BEGIN CERTIFICATE-----
  <MY_TRUSTED_CA_CERT>
  -----END CERTIFICATE-----
additionalTrustBundlePolicy: <policy_to_add_additionalTrustBundle>
# ...

```

where:

proxy.httpProxy

Specifies a proxy URL to use for creating HTTP connections outside the cluster. The URL scheme must be **http**.

proxy.httpsProxy

Specifies a proxy URL to use for creating HTTPS connections outside the cluster.

proxy.noProxy

Specifies a comma-separated list of destination domain names, IP addresses, or other network CIDRs to exclude from proxying. Preface a domain with `.` to match subdomains only. For example, `.y.com` matches `x.y.com`, but not `y.com`. Use `*` to bypass the proxy for all destinations. You must include vCenter's IP address and the IP range that you use for its machines.

additionalTrustBundle

If provided, the installation program generates a config map that is named **user-ca-bundle** in the **openshift-config** namespace to hold the additional CA certificates. If you provide **additionalTrustBundle** and at least one proxy setting, the **Proxy** object is configured to reference the **user-ca-bundle** config map in the **trustedCA** field. The Cluster Network Operator then creates a **trusted-ca-bundle** config map that merges the contents specified for the **trustedCA** parameter with the RHCOS trust bundle. The **additionalTrustBundle** field is required unless the proxy's identity certificate is signed by an authority from the RHCOS trust bundle.

additionalTrustBundlePolicy

Specifies the policy that determines the configuration of the **Proxy** object to reference the **user-ca-bundle** config map in the **trustedCA** field. The allowed values are **Proxyonly** and **Always**. Use **Proxyonly** to reference the **user-ca-bundle** config map only when **http/https** proxy is configured. Use **Always** to always reference the **user-ca-bundle** config map. The default value is **Proxyonly**. Optional parameter.



NOTE

The installation program does not support the proxy **readinessEndpoints** field.



NOTE

If the installer times out, restart and then complete the deployment by using the **wait-for** command of the installer. For example:

+

```
$ ./openshift-install wait-for install-complete --log-level debug
```

2. Save the file and reference it when installing OpenShift Container Platform.

The installation program creates a cluster-wide proxy that is named **cluster** that uses the proxy settings in the provided **install-config.yaml** file. If no proxy settings are provided, a **cluster Proxy** object is still created, but it will have a nil **spec**.



NOTE

Only the **Proxy** object named **cluster** is supported, and no additional proxies can be created.

2.4.5.3. Configuring regions and zones for a VMware vCenter

You can modify the default installation configuration file, so that you can deploy an OpenShift Container Platform cluster to multiple vSphere data centers.

The default **install-config.yaml** file configuration from the previous release of OpenShift Container Platform is deprecated. You can continue to use the deprecated default configuration, but the **openshift-installer** will prompt you with a warning message that indicates the use of deprecated fields in the configuration file.

Prerequisites

- You have an existing **install-config.yaml** installation configuration file.



IMPORTANT

You must specify at least one failure domain for your OpenShift Container Platform cluster, so that you can provision data center objects for your VMware vCenter server. Consider specifying multiple failure domains if you need to provision virtual machine nodes in different data centers, clusters, datastores, and other components. To enable regions and zones, you must define multiple failure domains for your OpenShift Container Platform cluster.

- You have installed the **govc** command line tool.



IMPORTANT

The example uses the **govc** command. The **govc** command is an open source command available from VMware; it is not available from Red Hat. The Red Hat support team does not maintain the **govc** command. Instructions for downloading and installing **govc** are found on the VMware documentation website.

Procedure

1. Create the **openshift-region** and **openshift-zone** vCenter tag categories by running the following commands:



IMPORTANT

If you specify different names for the **openshift-region** and **openshift-zone** vCenter tag categories, the installation of the OpenShift Container Platform cluster fails.

```
$ govc tags.category.create -d "OpenShift region" openshift-region
```

```
$ govc tags.category.create -d "OpenShift zone" openshift-zone
```

2. For each region where you want to deploy your cluster, create a region tag by running the following command:

```
$ govc tags.create -c <region_tag_category> <region_tag>
```

3. For each zone where you want to deploy your cluster, create a zone tag by running the following command:

```
$ govc tags.create -c <zone_tag_category> <zone_tag>
```

4. Attach region tags to each vCenter data center object by running the following command:

```
$ govc tags.attach -c <region_tag_category> <region_tag_1> /<data_center_1>
```

5. Attach the zone tags to each vCenter cluster object by running the following command:

```
$ govc tags.attach -c <zone_tag_category> <zone_tag_1> /<data_center_1>/host/<cluster1>
```

6. Change to the directory that contains the installation program and initialize the cluster deployment according to your chosen installation requirements.

Sample `install-config.yaml` file with multiple data centers defined in a vSphere center

```
# ...
compute:
---
vsphere:
  zones:
    - "<machine_pool_zone_1>"
    - "<machine_pool_zone_2>"
# ...
controlPlane:
# ...
vsphere:
  zones:
    - "<machine_pool_zone_1>"
    - "<machine_pool_zone_2>"
```

```

# ...
platform:
  vsphere:
    vcenters:
# ...
  datacenters:
    - <data_center_1_name>
    - <data_center_2_name>
  failureDomains:
    - name: <machine_pool_zone_1>
      region: <region_tag_1>
      zone: <zone_tag_1>
      server: <fully_qualified_domain_name>
      topology:
        datacenter: <data_center_1>
        computeCluster: "/<data_center_1>/host/<cluster1>"
        networks:
          - <VM_Network1_name>
          datastore: "/<data_center_1>/datastore/<datastore1>"
          resourcePool: "/<data_center_1>/host/<cluster1>/Resources/<resourcePool1>"
          folder: "/<data_center_1>/vm/<folder1>"
    - name: <machine_pool_zone_2>
      region: <region_tag_2>
      zone: <zone_tag_2>
      server: <fully_qualified_domain_name>
      topology:
        datacenter: <data_center_2>
        computeCluster: "/<data_center_2>/host/<cluster2>"
        networks:
          - <VM_Network2_name>
          datastore: "/<data_center_2>/datastore/<datastore2>"
          resourcePool: "/<data_center_2>/host/<cluster2>/Resources/<resourcePool2>"
          folder: "/<data_center_2>/vm/<folder2>"
# ...

```

2.4.5.4. Configuring host groups for a VMware vCenter



IMPORTANT

OpenShift zones support for vSphere host groups is a Technology Preview feature only. Technology Preview features are not supported with Red Hat production service level agreements (SLAs) and might not be functionally complete. Red Hat does not recommend using them in production. These features provide early access to upcoming product features, enabling customers to test functionality and provide feedback during the development process.

For more information about the support scope of Red Hat Technology Preview features, see [Technology Preview Features Support Scope](#).

You can modify the default installation configuration file to deploy an OpenShift Container Platform cluster on a VMware vSphere stretched cluster, where ESXi hosts are grouped into host groups by physical location.

The default **install-config.yaml** file configuration from previous releases of OpenShift Container Platform is deprecated. Though you can still use it, the OpenShift Container Platform installer will display a warning message that indicates the use of deprecated fields in the configuration file.

Prerequisites

- You have an existing **install-config.yaml** installation configuration file.
- You have arranged your ESXi hosts into host groups.
- You have granted the **Host.Inventory.EditCluster** privilege on the vSphere vCenter cluster object.
- You have downloaded and installed the **govc** command line tool. Instructions can be found on the VMware documentation website. Note that **govc** is an open-source tool that is not maintained by the Red Hat support team.
- You have enabled the **TechPreviewNoUpgrade** feature set. For more information, see "Enabling features using feature gates".



IMPORTANT

To enable host group support, you must define multiple failure domains for your OpenShift Container Platform cluster.

Procedure

1. Create the **openshift-region** and **openshift-zone** vCenter tag categories by running the following commands:



IMPORTANT

If you specify different names for the **openshift-region** and **openshift-zone** vCenter tag categories, the installation of the OpenShift Container Platform cluster fails.

```
$ govc tags.category.create -d "OpenShift region" openshift-region
```

```
$ govc tags.category.create -d "OpenShift zone" openshift-zone
```

2. Create a region tag for the vSphere cluster where you want to deploy your OpenShift Container Platform cluster by entering the following command:

```
$ govc tags.create -c <region_tag_category> <region_tag>
```

3. Create a zone tag for each host group by entering the following command as needed:

```
$ govc tags.create -c <zone_tag_category> <zone_tag>
```

4. Attach the region tag to the vCenter cluster object by entering the following command:

```
$ govc tags.attach -c <region_tag_category> <region_tag_1>
/<datacenter_1>/host/<cluster_1>
```

5. Use zone tags to associate each ESXi host with its host group, by entering the following command for each ESXi host:

```
$ govc tags.attach -c <zone_tag_category> <zone_tag_for_host_group_1>
/<datacenter_1>/host/<cluster_1>/<esxi_host_in_host_group_1>
```

6. Change to the directory that contains the installation program and initialize the cluster deployment according to your chosen installation requirements.

Sample `install-config.yaml` file with multiple host groups

```
featureSet: TechPreviewNoUpgrade
featureGate:
  - "VSphereHostVMGroupZonal=true"
# ...
platform:
  vsphere:
    vcenters:
# ...
    datacenters:
      - <data_center_1_name>
failureDomains:
  - name: <host_group_1>
    region: <cluster_1_region_tag>
    zone: <host_group_1_zone_tag>
    regionType: "ComputeCluster"
    zoneType: "HostGroup"
    server: <fully_qualified_domain_name>
    topology:
      datacenter: <data_center_1>
      computeCluster: "/<data_center_1>/host/<cluster_1>"
      networks:
        - <VM_Network1_name>
      hostGroup: <host_group_1_name>
      datastore: "/<data_center_1>/datastore/<datastore_1>"
      resourcePool: "/<data_center_1>/host/<cluster_1>/Resources/<resourcePool_1>"
      folder: "/<data_center_1>/vm/<folder_1>"
  - name: <host_group_2>
    region: <cluster_1_region_tag>
    zone: <host_group_2_zone_tag>
    regionType: "ComputeCluster"
    zoneType: "HostGroup"
    server: <fully_qualified_domain_name>
    topology:
      datacenter: <data_center_1>
      computeCluster: "/<data_center_1>/host/<cluster_1>"
      networks:
        - <VM_Network1_name>
      hostGroup: <host_group_2_name>
      datastore: "/<data_center_1>/datastore/<datastore_1>"
      resourcePool: "/<data_center_1>/host/<cluster_1>/Resources/<resourcePool_1>"
      folder: "/<data_center_1>/vm/<folder_1>"
```

Additional resources

- [Enabling features using feature gates](#)

2.4.6. Services for a user-managed load balancer

To integrate your infrastructure with existing network standards or gain more control over traffic management in OpenShift Container Platform, configure services for a user-managed load balancer.



IMPORTANT

Configuring a user-managed load balancer depends on your vendor's load balancer.

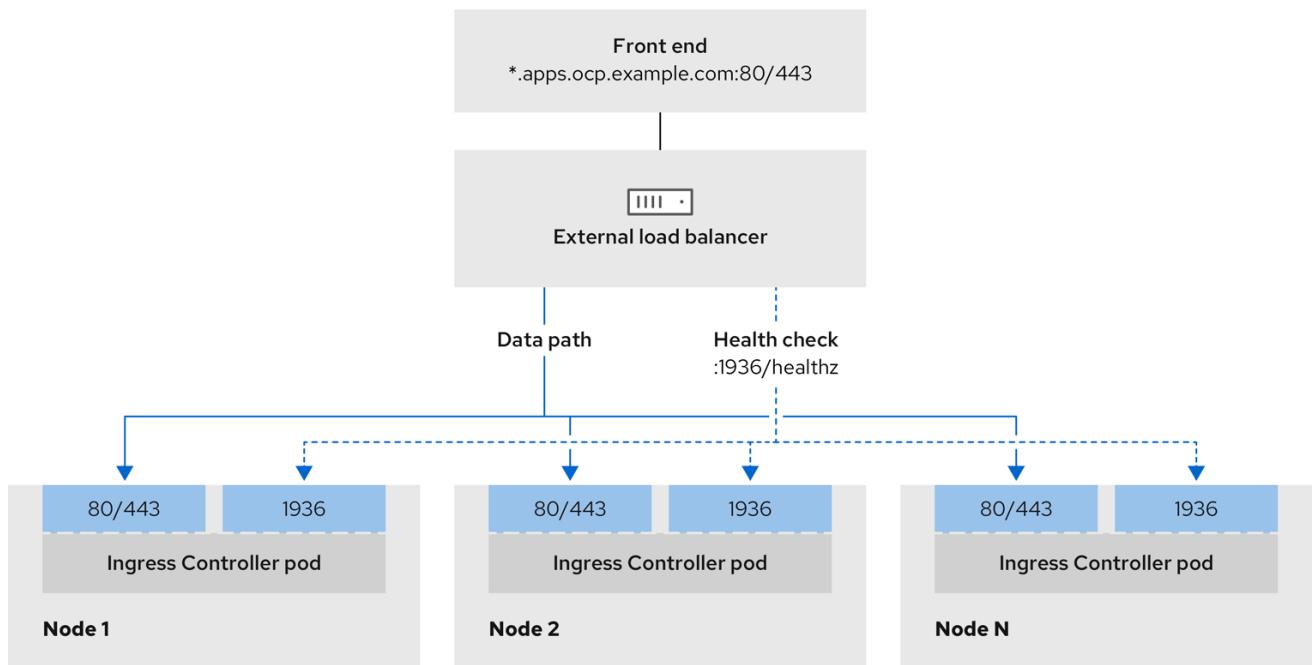
The information and examples in this section are for guideline purposes only. Consult the vendor documentation for more specific information about the vendor's load balancer.

Red Hat supports the following services for a user-managed load balancer:

- Ingress Controller
- OpenShift API
- OpenShift MachineConfig API

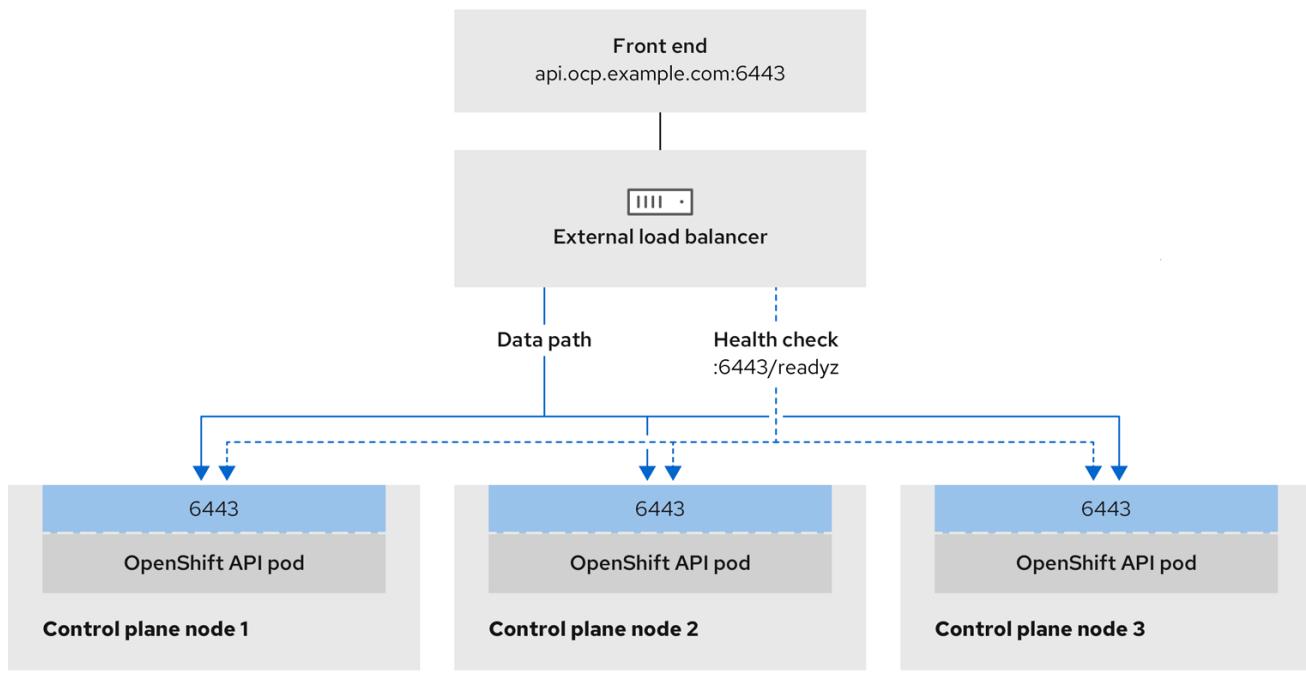
You can choose whether you want to configure one or all of these services for a user-managed load balancer. Configuring only the Ingress Controller service is a common configuration option. To better understand each service, view the following diagrams:

Figure 2.1. Example network workflow that shows an Ingress Controller operating in an OpenShift Container Platform environment



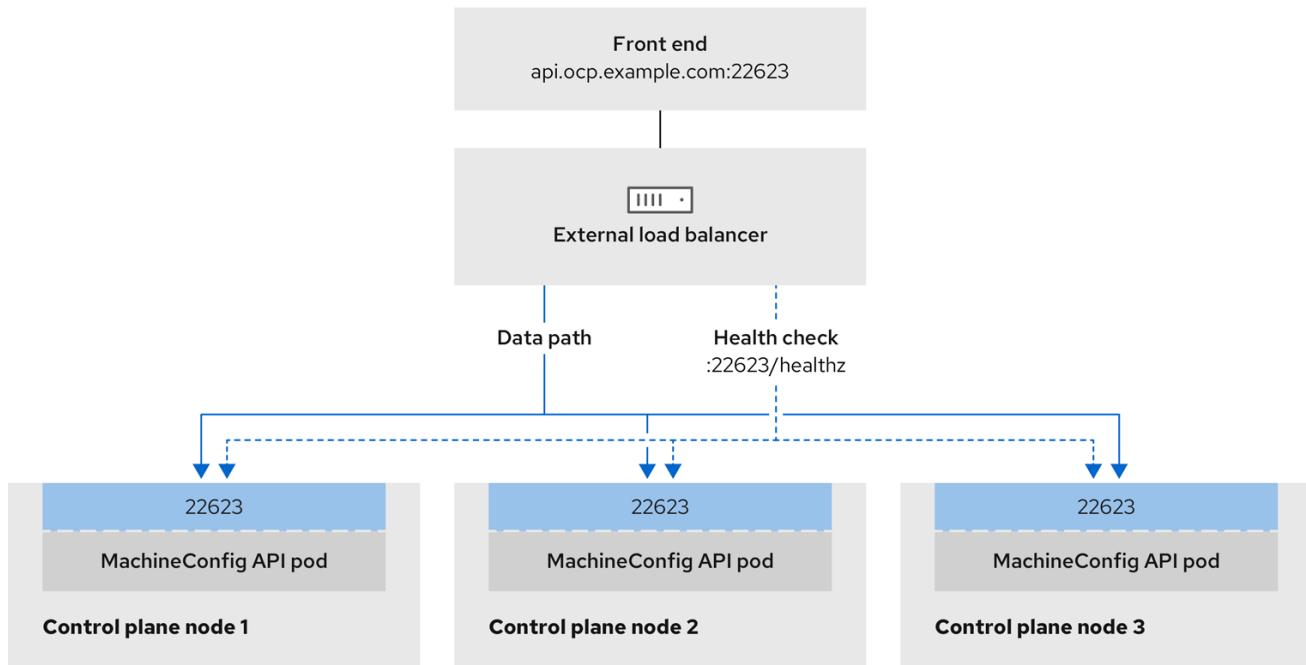
496_OpenShift_1223

Figure 2.2. Example network workflow that shows an OpenShift API operating in an OpenShift Container Platform environment



496_OpenShift_I223

Figure 2.3. Example network workflow that shows an OpenShift MachineConfig API operating in an OpenShift Container Platform environment



496_OpenShift_I223

The following configuration options are supported for user-managed load balancers:

- Use a node selector to map the Ingress Controller to a specific set of nodes. You must assign a static IP address to each node in this set, or configure each node to receive the same IP address from the Dynamic Host Configuration Protocol (DHCP). Infrastructure nodes commonly receive this type of configuration.

- Target all IP addresses on a subnet. This configuration can reduce maintenance overhead, because you can create and destroy nodes within those networks without reconfiguring the load balancer targets. If you deploy your ingress pods by using a machine set on a smaller network, such as a /27 or /28, you can simplify your load balancer targets.

TIP

You can list all IP addresses that exist in a network by checking the machine config pool's resources.

Before you configure a user-managed load balancer for your OpenShift Container Platform cluster, consider the following information:

- For a front-end IP address, you can use the same IP address for the front-end IP address, the Ingress Controller's load balancer, and API load balancer. Check the vendor's documentation for this capability.
- For a back-end IP address, ensure that an IP address for an OpenShift Container Platform control plane node does not change during the lifetime of the user-managed load balancer. You can achieve this by completing one of the following actions:
 - Assign a static IP address to each control plane node.
 - Configure each node to receive the same IP address from the DHCP every time the node requests a DHCP lease. Depending on the vendor, the DHCP lease might be in the form of an IP reservation or a static DHCP assignment.
- Manually define each node that runs the Ingress Controller in the user-managed load balancer for the Ingress Controller back-end service. For example, if the Ingress Controller moves to an undefined node, a connection outage can occur.

2.4.6.1. Configuring a user-managed load balancer

To integrate your infrastructure with existing network standards or gain more control over traffic management in OpenShift Container Platform, use a user-managed load balancer in place of the default load balancer.



IMPORTANT

Before you configure a user-managed load balancer, ensure that you read the "Services for a user-managed load balancer" section.

Read the following prerequisites that apply to the service that you want to configure for your user-managed load balancer.



NOTE

MetalLB, which runs on a cluster, functions as a user-managed load balancer.

Prerequisites

The following list details OpenShift API prerequisites:

- You defined a front-end IP address.

- TCP ports 6443 and 22623 are exposed on the front-end IP address of your load balancer. Check the following items:
 - Port 6443 provides access to the OpenShift API service.
 - Port 22623 can provide ignition startup configurations to nodes.
- The front-end IP address and port 6443 are reachable by all users of your system with a location external to your OpenShift Container Platform cluster.
- The front-end IP address and port 22623 are reachable only by OpenShift Container Platform nodes.
- The load balancer backend can communicate with OpenShift Container Platform control plane nodes on port 6443 and 22623.

The following list details Ingress Controller prerequisites:

- You defined a front-end IP address.
- TCP port 443 and port 80 are exposed on the front-end IP address of your load balancer.
- The front-end IP address, port 80 and port 443 are reachable by all users of your system with a location external to your OpenShift Container Platform cluster.
- The front-end IP address, port 80 and port 443 are reachable by all nodes that operate in your OpenShift Container Platform cluster.
- The load balancer backend can communicate with OpenShift Container Platform nodes that run the Ingress Controller on ports 80, 443, and 1936.

The following list details prerequisites for health check URL specifications:

You can configure most load balancers by setting health check URLs that determine if a service is available or unavailable. OpenShift Container Platform provides these health checks for the OpenShift API, Machine Configuration API, and Ingress Controller backend services.

The following example shows a Kubernetes API health check specification for a backend service:

```
Path: HTTPS:6443/readyz
Healthy threshold: 2
Unhealthy threshold: 2
Timeout: 10
Interval: 10
```

The following example shows a Machine Config API health check specification for a backend service:

```
Path: HTTPS:22623/healthz
Healthy threshold: 2
Unhealthy threshold: 2
Timeout: 10
Interval: 10
```

The following example shows a Ingress Controller health check specification for a backend service:

```
Path: HTTP:1936/healthz/ready
```

Healthy threshold: 2
 Unhealthy threshold: 2
 Timeout: 5
 Interval: 10

Procedure

1. Configure the HAProxy Ingress Controller, so that you can enable access to the cluster from your load balancer on ports 6443, 22623, 443, and 80. Depending on your needs, you can specify the IP address of a single subnet or IP addresses from multiple subnets in your HAProxy configuration.

Example HAProxy configuration with one listed subnet

```
# ...
listen my-cluster-api-6443
  bind 192.168.1.100:6443
  mode tcp
  balance roundrobin
  option httpchk
  http-check connect
  http-check send meth GET uri /readyz
  http-check expect status 200
  server my-cluster-master-2 192.168.1.101:6443 check inter 10s rise 2 fall 2
  server my-cluster-master-0 192.168.1.102:6443 check inter 10s rise 2 fall 2
  server my-cluster-master-1 192.168.1.103:6443 check inter 10s rise 2 fall 2

listen my-cluster-machine-config-api-22623
  bind 192.168.1.100:22623
  mode tcp
  balance roundrobin
  option httpchk
  http-check connect
  http-check send meth GET uri /healthz
  http-check expect status 200
  server my-cluster-master-2 192.168.1.101:22623 check inter 10s rise 2 fall 2
  server my-cluster-master-0 192.168.1.102:22623 check inter 10s rise 2 fall 2
  server my-cluster-master-1 192.168.1.103:22623 check inter 10s rise 2 fall 2

listen my-cluster-apps-443
  bind 192.168.1.100:443
  mode tcp
  balance roundrobin
  option httpchk
  http-check connect
  http-check send meth GET uri /healthz/ready
  http-check expect status 200
  server my-cluster-worker-0 192.168.1.111:443 check port 1936 inter 10s rise 2 fall 2
  server my-cluster-worker-1 192.168.1.112:443 check port 1936 inter 10s rise 2 fall 2
  server my-cluster-worker-2 192.168.1.113:443 check port 1936 inter 10s rise 2 fall 2

listen my-cluster-apps-80
  bind 192.168.1.100:80
  mode tcp
  balance roundrobin
  option httpchk
```

```

http-check connect
http-check send meth GET uri /healthz/ready
http-check expect status 200
  server my-cluster-worker-0 192.168.1.111:80 check port 1936 inter 10s rise 2 fall 2
  server my-cluster-worker-1 192.168.1.112:80 check port 1936 inter 10s rise 2 fall 2
  server my-cluster-worker-2 192.168.1.113:80 check port 1936 inter 10s rise 2 fall 2
# ...

```

Example HAProxy configuration with multiple listed subnets

```

# ...
listen api-server-6443
  bind *:6443
  mode tcp
    server master-00 192.168.83.89:6443 check inter 1s
    server master-01 192.168.84.90:6443 check inter 1s
    server master-02 192.168.85.99:6443 check inter 1s
    server bootstrap 192.168.80.89:6443 check inter 1s

listen machine-config-server-22623
  bind *:22623
  mode tcp
    server master-00 192.168.83.89:22623 check inter 1s
    server master-01 192.168.84.90:22623 check inter 1s
    server master-02 192.168.85.99:22623 check inter 1s
    server bootstrap 192.168.80.89:22623 check inter 1s

listen ingress-router-80
  bind *:80
  mode tcp
  balance source
    server worker-00 192.168.83.100:80 check inter 1s
    server worker-01 192.168.83.101:80 check inter 1s

listen ingress-router-443
  bind *:443
  mode tcp
  balance source
    server worker-00 192.168.83.100:443 check inter 1s
    server worker-01 192.168.83.101:443 check inter 1s

listen ironic-api-6385
  bind *:6385
  mode tcp
  balance source
    server master-00 192.168.83.89:6385 check inter 1s
    server master-01 192.168.84.90:6385 check inter 1s
    server master-02 192.168.85.99:6385 check inter 1s
    server bootstrap 192.168.80.89:6385 check inter 1s

listen inspector-api-5050
  bind *:5050
  mode tcp
  balance source
    server master-00 192.168.83.89:5050 check inter 1s
    server master-01 192.168.84.90:5050 check inter 1s

```

```
server master-02 192.168.85.99:5050 check inter 1s
server bootstrap 192.168.80.89:5050 check inter 1s
# ...
```

2. Use the **curl** CLI command to verify that the user-managed load balancer and its resources are operational:

a. Verify that the cluster machine configuration API is accessible to the Kubernetes API server resource, by running the following command and observing the response:

```
$ curl https://<loadbalancer_ip_address>:6443/version --insecure
```

If the configuration is correct, you receive a JSON object in response:

```
{
  "major": "1",
  "minor": "11+",
  "gitVersion": "v1.11.0+ad103ed",
  "gitCommit": "ad103ed",
  "gitTreeState": "clean",
  "buildDate": "2019-01-09T06:44:10Z",
  "goVersion": "go1.10.3",
  "compiler": "gc",
  "platform": "linux/amd64"
}
```

b. Verify that the cluster machine configuration API is accessible to the Machine config server resource, by running the following command and observing the output:

```
$ curl -v https://<loadbalancer_ip_address>:22623/healthz --insecure
```

If the configuration is correct, the output from the command shows the following response:

```
HTTP/1.1 200 OK
Content-Length: 0
```

c. Verify that the controller is accessible to the Ingress Controller resource on port 80, by running the following command and observing the output:

```
$ curl -I -L -H "Host: console-openshift-console.apps.<cluster_name>.<base_domain>" http://<load_balancer_front_end_IP_address>
```

If the configuration is correct, the output from the command shows the following response:

```
HTTP/1.1 302 Found
content-length: 0
location: https://console-openshift-console.apps.ocp4.private.opequon.net/
cache-control: no-cache
```

d. Verify that the controller is accessible to the Ingress Controller resource on port 443, by running the following command and observing the output:

```
$ curl -I -L --insecure --resolve console-openshift-console.apps.<cluster_name>.<base_domain>:443:<Load Balancer Front End IP Address> https://console-openshift-console.apps.<cluster_name>.<base_domain>
```

If the configuration is correct, the output from the command shows the following response:

```
HTTP/1.1 200 OK
referrer-policy: strict-origin-when-cross-origin
set-cookie: csrf-
token=UIYWOyQ62LWjw2h003xtYSKlh1a0Py2hhctw0WmV2YEdhJjFyQwWcGBsja261dG
LgaYO0nxzVERhiXt6QepA7g==; Path=/; Secure; SameSite=Lax
x-content-type-options: nosniff
x-dns-prefetch-control: off
x-frame-options: DENY
x-xss-protection: 1; mode=block
date: Wed, 04 Oct 2023 16:29:38 GMT
content-type: text/html; charset=utf-8
set-cookie:
1e2670d92730b515ce3a1bb65da45062=1bf5e9573c9a2760c964ed1659cc1673; path=/
HttpOnly; Secure; SameSite=None
cache-control: private
```

3. Configure the DNS records for your cluster to target the front-end IP addresses of the user-managed load balancer. You must update records to your DNS server for the cluster API and applications over the load balancer. The following examples shows modified DNS records:

<load_balancer_ip_address> [A](#) api.<cluster_name>.<base_domain>
 A record pointing to Load Balancer Front End

<load_balancer_ip_address> [A](#) apps.<cluster_name>.<base_domain>
 A record pointing to Load Balancer Front End



IMPORTANT

DNS propagation might take some time for each DNS record to become available. Ensure that each DNS record propagates before validating each record.

4. For your OpenShift Container Platform cluster to use the user-managed load balancer, you must specify the following configuration in your cluster's **install-config.yaml** file:

```
# ...
platform:
  vsphere:
    loadBalancer:
      type: UserManaged
    apiVIPs:
      - <api_ip> 1
    ingressVIPs:
      - <ingress_ip> 2
# ...
```

where:

loadBalancer.type

Set **UserManaged** for the **type** parameter to specify a user-managed load balancer for your cluster. The parameter defaults to **OpenShiftManagedDefault**, which denotes the default internal load balancer. For services defined in an **openshift-kni-infra** namespace, a user-managed load balancer can deploy the **coredns** service to pods in your cluster but ignores **keepalived** and **haproxy** services.

loadBalancer.<api_ip>

Specifies a user-managed load balancer. Specify the user-managed load balancer's public IP address, so that the Kubernetes API can communicate with the user-managed load balancer. Mandatory parameter.

loadBalancer.<ingress_ip>

Specifies a user-managed load balancer. Specify the user-managed load balancer's public IP address, so that the user-managed load balancer can manage ingress traffic for your cluster. Mandatory parameter.

Verification

1. Use the **curl** CLI command to verify that the user-managed load balancer and DNS record configuration are operational:
 - a. Verify that you can access the cluster API, by running the following command and observing the output:

```
$ curl https://api.<cluster_name>.<base_domain>:6443/version --insecure
```

If the configuration is correct, you receive a JSON object in response:

```
{
  "major": "1",
  "minor": "11+",
  "gitVersion": "v1.11.0+ad103ed",
  "gitCommit": "ad103ed",
  "gitTreeState": "clean",
  "buildDate": "2019-01-09T06:44:10Z",
  "goVersion": "go1.10.3",
  "compiler": "gc",
  "platform": "linux/amd64"
}
```

- b. Verify that you can access the cluster machine configuration, by running the following command and observing the output:

```
$ curl -v https://api.<cluster_name>.<base_domain>:22623/healthz --insecure
```

If the configuration is correct, the output from the command shows the following response:

```
HTTP/1.1 200 OK
Content-Length: 0
```

- c. Verify that you can access each cluster application on port 80, by running the following command and observing the output:

```
$ curl http://console-openshift-console.apps.<cluster_name>.<base_domain> -I -L --insecure
```

If the configuration is correct, the output from the command shows the following response:

```
HTTP/1.1 302 Found
content-length: 0
location: https://console-openshift-console.apps.<cluster-name>.<base domain>/
cache-control: no-cacheHTTP/1.1 200 OK
referrer-policy: strict-origin-when-cross-origin
set-cookie: csrf-
token=39HoZgztDnzjJkq/JuLJMeoKNXIfiVv2YgZc09c3TBOBU4NI6kDXaJH1LdicNhN1UsQ
Wzon4Dor9GWGfopaTEQ==; Path=/; Secure
x-content-type-options: nosniff
x-dns-prefetch-control: off
x-frame-options: DENY
x-xss-protection: 1; mode=block
date: Tue, 17 Nov 2020 08:42:10 GMT
content-type: text/html; charset=utf-8
set-cookie:
1e2670d92730b515ce3a1bb65da45062=9b714eb87e93cf34853e87a92d6894be; path=/;
HttpOnly; Secure; SameSite=None
cache-control: private
```

d. Verify that you can access each cluster application on port 443, by running the following command and observing the output:

```
$ curl https://console-openshift-console.apps.<cluster_name>.<base_domain> -I -L --insecure
```

If the configuration is correct, the output from the command shows the following response:

```
HTTP/1.1 200 OK
referrer-policy: strict-origin-when-cross-origin
set-cookie: csrf-
token=UIYW0yQ62LWjw2h003xtYSKlh1a0Py2hhctw0WmV2YEdhJjFyQwWcGBsja261dG
LgaYO0nxzVErhiXt6QepA7g==; Path=/; Secure; SameSite=Lax
x-content-type-options: nosniff
x-dns-prefetch-control: off
x-frame-options: DENY
x-xss-protection: 1; mode=block
date: Wed, 04 Oct 2023 16:29:38 GMT
content-type: text/html; charset=utf-8
set-cookie:
1e2670d92730b515ce3a1bb65da45062=1bf5e9573c9a2760c964ed1659cc1673; path=/;
HttpOnly; Secure; SameSite=None
cache-control: private
```

2.4.7. Deploying 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

- You have the OpenShift Container Platform installation program and the pull secret for your cluster.
- You have verified that the cloud provider account on your host has the correct permissions to deploy the cluster. An account with incorrect permissions causes the installation process to fail with an error message that displays the missing permissions.
- Optional: Before you create the cluster, you configured an external load balancer in place of the default load balancer.



IMPORTANT

You do not need to specify API and Ingress static addresses for your installation program. If you choose this configuration, you must take additional actions to define network targets that accept an IP address from each referenced vSphere subnet. See the section "Configuring a user-managed load balancer".

Procedure

- In the directory that contains the installation program, initialize the cluster deployment by running the following command:

```
$ ./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**.

Verification

When the cluster deployment completes successfully:

- The terminal displays directions for accessing your cluster, including a link to the web console and credentials for the **kubeadmin** user.
- Credential information also outputs to **<installation_directory>/openshift_install.log**.



IMPORTANT

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

Example output

```
...
INFO Install complete!
INFO To access the cluster as the system:admin user when using 'oc', run 'export
KUBECONFIG=/home/myuser/install_dir/auth/kubeconfig'
INFO Access the OpenShift web-console here: https://console-openshift-
console.apps.mycluster.example.com
INFO Login to the console with user: "kubeadmin", and password: "password"
INFO Time elapsed: 36m22s
```



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.
- It is recommended that you use 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.

2.4.8. Logging in to the cluster by using the CLI

To log in to your cluster as the default system user, export the **kubeconfig** file. This configuration enables the CLI to authenticate and connect to the specific API server created during OpenShift Container Platform installation.

The **kubeconfig** file is specific to a cluster and is created during OpenShift Container Platform installation.

Prerequisites

- You deployed an OpenShift Container Platform cluster.
- You installed the OpenShift CLI (**oc**).

Procedure

1. Export the **kubeadm** credentials by running the following command:

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

where:

<installation_directory>

Specifies the path to the directory that stores the installation files.

2. Verify you can run **oc** commands successfully using the exported configuration by running the following command:

■

```
$ oc whoami
```

Example output

```
system:admin
```

2.4.9. Creating registry storage

After you install the cluster, you must create storage for the registry Operator.

2.4.9.1. Image registry removed during installation

On platforms that do not provide shareable object storage, the OpenShift Image Registry Operator bootstraps itself as **Removed**. This allows **openshift-installer** to complete installations on these platform types.

After installation, you must edit the Image Registry Operator configuration to switch the **managementState** from **Removed** to **Managed**. When this has completed, you must configure storage.

2.4.9.2. Image registry storage configuration

The Image Registry Operator is not initially available for platforms that do not provide default storage. After installation, you must configure your registry to use storage so that the Registry Operator is made available.

Configure a persistent volume, which is required for production clusters. Where applicable, you can configure an empty directory as the storage location for non-production clusters.

You can also allow the image registry to use block storage types by using the **Recreate** rollout strategy during upgrades.

2.4.9.2.1. Configuring registry storage for VMware vSphere

As a cluster administrator, following installation you must configure your registry to use storage.

Prerequisites

- Cluster administrator permissions.
- A cluster on VMware vSphere.
- Persistent storage provisioned for your cluster, such as Red Hat OpenShift Data Foundation.



IMPORTANT

OpenShift Container Platform supports **ReadWriteOnce** access for image registry storage when you have only one replica. **ReadWriteOnce** access also requires that the registry uses the **Recreate** rollout strategy. To deploy an image registry that supports high availability with two or more replicas, **ReadWriteMany** access is required.

- Must have "100Gi" capacity.



IMPORTANT

Testing shows issues with using the NFS server on RHEL as storage backend for core services. This includes the OpenShift Container Registry and Quay, Prometheus for monitoring storage, and Elasticsearch for logging storage. Therefore, using RHEL NFS to back PVs used by core services is not recommended.

Other NFS implementations on the marketplace might not have these issues. Contact the individual NFS implementation vendor for more information on any testing that was possibly completed against these OpenShift Container Platform core components.

Procedure

1. Change the **spec.storage.pvc** field in the **configs.imageregistry/cluster** resource.



NOTE

When you use shared storage, review your security settings to prevent outside access.

2. Verify that you do not have a registry pod by running the following command:

```
$ oc get pod -n openshift-image-registry -l docker-registry=default
```

Example output

```
No resources found in openshift-image-registry namespace
```



NOTE

If you do have a registry pod in your output, you do not need to continue with this procedure.

3. Check the registry configuration by running the following command:

```
$ oc edit configs.imageregistry.operator.openshift.io
```

Example output

```
storage:
  pvc:
    claim:
```

Leave the **claim** field blank to allow the automatic creation of an **image-registry-storage** persistent volume claim (PVC). The PVC is generated based on the default storage class. However, be aware that the default storage class might provide ReadWriteOnce (RWO) volumes, such as a RADOS Block Device (RBD), which can cause issues when you replicate to more than one replica.

4. Check the **clusteroperator** status by running the following command:

```
$ oc get clusteroperator image-registry
```

Example output

NAME	VERSION	AVAILABLE	PROGRESSING	DEGRADED	SINCE
MESSAGE					
image-registry	4.7	True	False	False	6h50m

2.4.9.2.2. Configuring block registry storage for VMware vSphere

To allow the image registry to use block storage types such as vSphere Virtual Machine Disk (VMDK) during upgrades as a cluster administrator, you can use the **Recreate** rollout strategy.



IMPORTANT

Block storage volumes are supported but not recommended for use with image registry on production clusters. An installation where the registry is configured on block storage is not highly available because the registry cannot have more than one replica.

Procedure

1. Enter the following command to set the image registry storage as a block storage type, patch the registry so that it uses the **Recreate** rollout strategy, and runs with only **1** replica:

```
$ oc patch config.imageregistry.operator.openshift.io/cluster --type=merge -p '{"spec": {"rolloutStrategy": "Recreate", "replicas": 1}}'
```

2. Provision the persistent volume (PV) for the block storage device, and create a persistent volume claim (PVC) for that volume. The requested block volume uses the `ReadWriteOnce` (RWO) access mode.
 - a. Create a **pvc.yaml** file with the following contents to define a VMware vSphere **PersistentVolumeClaim** object:

```
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: image-registry-storage
  namespace: openshift-image-registry
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 100Gi
```

where:

name

Specifies a unique name that represents the **PersistentVolumeClaim** object.

namespace

Specifies the **namespace** for the **PersistentVolumeClaim** object, which is **openshift-image-registry**.

accessModes

Specifies the access mode of the persistent volume claim. With **ReadWriteOnce**, the volume can be mounted with read and write permissions by a single node.

storage

The size of the persistent volume claim.

- b. Enter the following command to create the **PersistentVolumeClaim** object from the file:

```
$ oc create -f pvc.yaml -n openshift-image-registry
```

3. Enter the following command to edit the registry configuration so that it references the correct PVC:

```
$ oc edit config.imageregistry.operator.openshift.io -o yaml
```

Example output

```
storage:
pvc:
claim:
```

By creating a custom PVC, you can leave the **claim** field blank for the default automatic creation of an **image-registry-storage** PVC.

For instructions about configuring registry storage so that it references the correct PVC, see [Configuring the registry for vSphere](#).

2.4.10. Telemetry access for OpenShift Container Platform

To provide metrics about cluster health and the success of updates, the Telemetry service requires internet access. When connected, this service runs automatically by default and registers your cluster to [OpenShift Cluster Manager](#).

After you confirm that your [OpenShift Cluster Manager](#) inventory is correct, either maintained automatically by Telemetry or manually by using OpenShift Cluster Manager, use subscription watch to track your OpenShift Container Platform subscriptions at the account or multi-cluster level. For more information about subscription watch, see "Data Gathered and Used by Red Hat's subscription services" in the *Additional resources* section.

Additional resources

- See [About remote health monitoring](#) for more information about the Telemetry service

2.4.11. Next steps

- [Customize your cluster](#).
- If necessary, you can [Remote health reporting](#).
- [Set up your registry and configure registry storage](#).
- Optional: [View the events from the vSphere Problem Detector Operator](#) to determine if the cluster has permission or storage configuration issues.

2.5. INSTALLING A CLUSTER ON VSPHERE WITH NETWORK CUSTOMIZATIONS

In OpenShift Container Platform version 4.19, you can install a cluster on your VMware vSphere instance by using installer-provisioned infrastructure with customized network configuration options. By customizing your network configuration, your cluster can coexist with existing IP address allocations in your environment and integrate with existing MTU and VXLAN configurations. To customize the installation, you modify parameters in the **install-config.yaml** file before you install the cluster.

You must set most of the network configuration parameters during installation, and you can modify only **kubeProxy** configuration parameters in a running cluster.

2.5.1. Prerequisites

- You have completed the tasks in [Preparing to install a cluster using installer-provisioned infrastructure](#).
- You reviewed your VMware platform licenses. Red Hat does not place any restrictions on your VMware licenses, but some VMware infrastructure components require licensing.
- You reviewed details about the [OpenShift Container Platform installation and update](#) processes.
- You read the documentation on [selecting a cluster installation method and preparing it for users](#).
- You provisioned [persistent storage](#) for your cluster. To deploy a private image registry, your storage must provide **ReadWriteMany** access modes.
- The OpenShift Container Platform installer requires access to port 443 on the vCenter and ESXi hosts. You verified that port 443 is accessible.
- If you use a firewall, confirm with the administrator that port 443 is accessible. Control plane nodes must be able to reach vCenter and ESXi hosts on port 443 for the installation to succeed.
- If you use a firewall, you [configured it to allow the sites](#) that your cluster requires access to.



NOTE

Be sure to also review this site list if you are configuring a proxy.

2.5.2. Internet access for OpenShift Container Platform

In OpenShift Container Platform 4.19, you require access to the internet to install your cluster.

You must have internet access to perform the following actions:

- Access [OpenShift Cluster Manager](#) to download the installation program and perform subscription management. If the cluster has internet access and you do not disable Telemetry, that service automatically entitles your cluster.
- 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 required content and use it to populate a mirror registry with the installation packages. 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.

2.5.3. VMware vSphere region and zone enablement

You can deploy an OpenShift Container Platform cluster to multiple vSphere data centers. Each data center can run multiple clusters. This configuration reduces the risk of a hardware failure or network outage that can cause your cluster to fail. To enable regions and zones, you must define multiple failure domains for your OpenShift Container Platform cluster.



IMPORTANT

The VMware vSphere region and zone enablement feature requires the vSphere Container Storage Interface (CSI) driver as the default storage driver in the cluster. As a result, the feature is only available on a newly installed cluster.

For a cluster that was upgraded from a previous release, you must enable CSI automatic migration for the cluster. You can then configure multiple regions and zones for the upgraded cluster.

The default installation configuration deploys a cluster to a single vSphere data center. If you want to deploy a cluster to multiple vSphere data centers, you must create an installation configuration file that enables the region and zone feature.

The default **install-config.yaml** file includes **vcenters** and **failureDomains** fields, where you can specify multiple vSphere data centers and clusters for your OpenShift Container Platform cluster. You can leave these fields blank if you want to install an OpenShift Container Platform cluster in a vSphere environment that consists of single data center.

The following list describes terms associated with defining zones and regions for your cluster:

- Failure domain: Establishes the relationships between a region and zone. You define a failure domain by using vCenter objects, such as a **datastore** object. A failure domain defines the vCenter location for OpenShift Container Platform cluster nodes.
- Region: Specifies a vCenter data center. You define a region by using a tag from the **openshift-region** tag category.
- Zone: Specifies a vCenter cluster. You define a zone by using a tag from the **openshift-zone** tag category.



NOTE

If you plan on specifying more than one failure domain in your **install-config.yaml** file, you must create tag categories, zone tags, and region tags in advance of creating the configuration file.

You must create a vCenter tag for each vCenter data center, which represents a region. Additionally, you must create a vCenter tag for each cluster than runs in a data center, which represents a zone. After you create the tags, you must attach each tag to their respective data centers and clusters.

The following table outlines an example of the relationship among regions, zones, and tags for a configuration with multiple vSphere data centers running in a single VMware vCenter.

Data center (region)	Cluster (zone)	Tags
us-east	us-east-1	us-east-1a
		us-east-1b
	us-east-2	us-east-2a
		us-east-2b
us-west	us-west-1	us-west-1a
		us-west-1b
	us-west-2	us-west-2a
		us-west-2b

2.5.4. VMware vSphere host group enablement



IMPORTANT

OpenShift zones support for vSphere host groups is a Technology Preview feature only. Technology Preview features are not supported with Red Hat production service level agreements (SLAs) and might not be functionally complete. Red Hat does not recommend using them in production. These features provide early access to upcoming product features, enabling customers to test functionality and provide feedback during the development process.

For more information about the support scope of Red Hat Technology Preview features, see [Technology Preview Features Support Scope](#).

When deploying an OpenShift Container Platform cluster to VMware vSphere, you can map your vSphere host groups onto OpenShift Container Platform failure domains. This is useful if you are using a stretched cluster configuration, where ESXi hosts are grouped into host groups by physical location.

To enable this feature, you must meet the following requirements:

- You must arrange your ESXi hosts into host groups.
- You must create a vCenter tag in the **openshift-region** tag category for your cluster. After you create the tag, you must attach the tag to the cluster.

- You must create a vCenter tag in the **openshift-zone** tag category for each host group and then attach the correct tag to each ESXi host.
- You must define multiple failure domains for your OpenShift Container Platform cluster in the **install-config.yaml** file.
- You must grant the **Host.Inventory.EditCluster** privilege on the vSphere vCenter cluster object.
- You must include the following parameters in your **install-config.yaml** file to enable this Technology Preview feature:

```
featureSet: TechPreviewNoUpgrade
featureGate:
  - "VSphereHostVMGroupZonal=true"
```



NOTE

For further information on feature gates, see "Enabling features using feature gates".

Review the following key terms, which correspond to parameters in your **install-config.yaml** file that you must configure to enable this feature:

- Failure domain: Specifies the relationships between regions and zones in OpenShift Container Platform, and clusters and host groups in vSphere. You define a failure domain by using vCenter objects, such as a **datastore** object. A failure domain defines the vCenter location for OpenShift Container Platform cluster nodes.
- Region: Specifies a vCenter cluster. You define a region by using a tag from the **openshift-region** tag category.
- Zone: Specifies a vCenter host group. You define a zone by using a tag from the **openshift-zone** tag category.
- Region type: Specifies the **ComputeCluster** region type to enable this feature.
- Zone type: Specifies the **HostGroup** zone type to enable this feature.

Additional resources

- [Additional VMware vSphere configuration parameters](#)
- [Deprecated VMware vSphere configuration parameters](#)
- [vSphere automatic migration](#)
- [VMware vSphere CSI Driver Operator](#)
- [Enabling features using feature gates](#)

2.5.5. Creating the installation configuration file

You can customize the OpenShift Container Platform cluster you install on

VMware vSphere.

Prerequisites

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

Procedure

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

- a. Change to the directory that contains the installation program and run the following command:

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

- **<installation_directory>**: For **<installation_directory>**, specify the directory name to store the files that the installation program creates.
When specifying the directory:
 - Verify that the directory has the **execute** permission. This permission is required to run Terraform binaries under the installation directory.
 - Use an empty directory. Some installation assets, such as bootstrap X.509 certificates, have short expiration intervals, therefore 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, specify an SSH key that your **ssh-agent** process uses.

- ii. Select **vsphere** as the platform to target.
- iii. Specify the name of your vCenter instance.
- iv. Specify the user name and password for the vCenter account that has the required permissions to create the cluster.
The installation program connects to your vCenter instance.
- v. Select the data center in your vCenter instance to connect to.



NOTE

After you create the installation configuration file, you can modify the file to create a multiple vSphere data center environment. This means that you can deploy an OpenShift Container Platform cluster to multiple vSphere data centers. For more information about creating this environment, see the section named *VMware vSphere region and zone enablement*.

- vi. Select the default vCenter datastore to use.



WARNING

You can specify the path of any datastore that exists in a datastore cluster. By default, Storage Distributed Resource Scheduler (SDRS), which uses Storage vMotion, is automatically enabled for a datastore cluster. Red Hat does not support Storage vMotion, so you must disable Storage DRS to avoid data loss issues for your OpenShift Container Platform cluster.

You cannot specify more than one datastore path. If you must specify VMs across multiple datastores, use a **datastore** object to specify a failure domain in your cluster's **install-config.yaml** configuration file. For more information, see "VMware vSphere region and zone enablement".

- vii. Select the vCenter cluster to install the OpenShift Container Platform cluster in. The installation program uses the root resource pool of the vSphere cluster as the default resource pool.
- viii. Select the network in the vCenter instance that contains the virtual IP addresses and DNS records that you configured.
- ix. Enter the virtual IP address that you configured for control plane API access.
- x. Enter the virtual IP address that you configured for cluster ingress.
- xi. Enter the base domain. This base domain must be the same one that you used in the DNS records that you configured.
- xii. Enter a descriptive name for your cluster.
The cluster name you enter must match the cluster name you specified when configuring the DNS records.

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.

Additional resources

- [Installation configuration parameters](#)

2.5.5.1. Sample `install-config.yaml` file for an installer-provisioned VMware vSphere cluster

You can customize the **install-config.yaml** file to specify more details about your OpenShift Container Platform cluster's platform or change the values of the required parameters.

```
apiVersion: v1
baseDomain: example.com 1
compute: 2
- architecture: amd64
  name: <worker_node>
  platform: {}
  replicas: 3
controlPlane: 3
  architecture: amd64
  name: <parent_node>
  platform: {}
  replicas: 3
metadata:
  creationTimestamp: null
  name: test 4
networking:
  clusterNetwork:
  - cidr: 10.128.0.0/14
    hostPrefix: 23
  machineNetwork:
  - cidr: 10.0.0.0/16
  networkType: OVNKubernetes 5
  serviceNetwork:
  - 172.30.0.0/16
platform:
  vsphere: 6
    apiVIPs:
    - 10.0.0.1
    failureDomains: 7
    - name: <failure_domain_name>
      region: <default_region_name>
      server: <fully_qualified_domain_name>
    topology:
      computeCluster: "/<data_center>/host/<cluster>" 8
      datacenter: <data_center>
      datastore: "/<data_center>/datastore/<datastore>" 8
    networks:
    - <VM_Network_name>
    resourcePool: "/<data_center>/host/<cluster>/Resources/<resourcePool>" 9
    folder: "/<data_center_name>/vm/<folder_name>/<subfolder_name>"
```

```

tagIDs: 10
  - <tag_id> 11
zone: <default_zone_name>
ingressVIPs:
  - 10.0.0.2
vcenters:
  - datacenters:
    - <data_center>
    password: <password>
    port: 443
    server: <fully_qualified_domain_name>
    user: administrator@vsphere.local
    diskType: thin 12
    fips: false
    pullSecret: '{"auths": ...}'
    sshKey: 'ssh-ed25519 AAAA...'
```

- 1 The base domain of the cluster. All DNS records must be sub-domains of this base and include the cluster name.
- 2, 3 The **controlPlane** section is a single mapping, but the **compute** section is a sequence of mappings. To meet the requirements of the different data structures, the first line of the **compute** section must begin with a hyphen, -, and the first line of the **controlPlane** section must not. Only one control plane pool is used.
- 4 The cluster name that you specified in your DNS records.
- 6 Optional: Provides additional configuration for the machine pool parameters for the compute and control plane machines.



IMPORTANT

The VIPs, **apiVIP** and **ingressVIP**, must come from the same **networking.machineNetwork** segment. For **apiVIP** and for **ingressVIP**, if the **networking.machineNetwork** is **10.0.0.0/16** then API VIPs and Ingress VIPs must be in one of the **10.0.0.0/16** machine networks.

- 7 Establishes the relationships between a region and zone. You define a failure domain by using vCenter objects, such as a **datastore** object. A failure domain defines the vCenter location for OpenShift Container Platform cluster nodes.
- 8 The path to the vSphere datastore that holds virtual machine files, templates, and ISO images.



IMPORTANT

You can specify the path of any datastore that exists in a datastore cluster. By default, Storage vMotion is automatically enabled for a datastore cluster. Red Hat does not support Storage vMotion, so you must disable Storage vMotion to avoid data loss issues for your OpenShift Container Platform cluster.

If you must specify VMs across multiple datastores, use a **datastore** object to specify a failure domain in your cluster's **install-config.yaml** configuration file. For more information, see "VMware vSphere region and zone enablement".

- 9 Optional: Provides an existing resource pool for machine creation. If you do not specify a value, the installation program uses the root resource pool of the vSphere cluster.
- 10 Optional: Each VM created by OpenShift Container Platform is assigned a unique tag that is specific to the cluster. The assigned tag enables the installation program to identify and remove the associated VMs when a cluster is decommissioned. You can list up to ten additional tag IDs to be attached to the VMs provisioned by the installation program.
- 11 The ID of the tag to be associated by the installation program. For example, **urn:vmomi:InventoryServiceTag:208e713c-cae3-4b7f-918e-4051ca7d1f97:GLOBAL**. For more information about determining the tag ID, see the [vSphere Tags and Attributes documentation](#).
- 12 The vSphere disk provisioning method.
- 5 The cluster network plugin to install. The default value **OVNKubernetes** is the only supported value.



NOTE

In OpenShift Container Platform 4.12 and later, the **apiVIP** and **ingressVIP** configuration settings are deprecated. Instead, use a list format to enter values in the **apiVIPs** and **ingressVIPs** configuration settings.

2.5.5.2. Configuring the cluster-wide proxy during installation

To enable internet access in environments that deny direct connections, configure a cluster-wide proxy in the **install-config.yaml** file. This configuration ensures that the new OpenShift Container Platform cluster routes traffic through the specified HTTP or HTTPS proxy.

Prerequisites

- You have an existing **install-config.yaml** file.
- You have reviewed the sites that your cluster requires access to and determined whether any of them need to bypass the proxy. By default, all cluster egress traffic is proxied, including calls to hosting cloud provider APIs. You added sites to the **Proxy** object's **spec.noProxy** field to bypass the proxy if necessary.



NOTE

The **Proxy** object **status.noProxy** field is populated with the values of the **networking.machineNetwork[].cidr**, **networking.clusterNetwork[].cidr**, and **networking.serviceNetwork[]** fields from your installation configuration.

For installations on Amazon Web Services (AWS), Google Cloud, Microsoft Azure, and Red Hat OpenStack Platform (RHOSP), the **Proxy** object **status.noProxy** field is also populated with the instance metadata endpoint (**169.254.169.254**).

Procedure

1. Edit your **install-config.yaml** file and add the proxy settings. For example:

```

apiVersion: v1
baseDomain: my.domain.com
proxy:
  httpProxy: http://<username>:<pswd>@<ip>:<port>
  httpsProxy: https://<username>:<pswd>@<ip>:<port>
  noProxy: example.com
  additionalTrustBundle: |
    -----BEGIN CERTIFICATE-----
    <MY_TRUSTED_CA_CERT>
    -----END CERTIFICATE-----
additionalTrustBundlePolicy: <policy_to_add_additionalTrustBundle>
# ...

```

where:

proxy.httpProxy

Specifies a proxy URL to use for creating HTTP connections outside the cluster. The URL scheme must be **http**.

proxy.httpsProxy

Specifies a proxy URL to use for creating HTTPS connections outside the cluster.

proxy.noProxy

Specifies a comma-separated list of destination domain names, IP addresses, or other network CIDRs to exclude from proxying. Preface a domain with `.` to match subdomains only. For example, `.y.com` matches `x.y.com`, but not `y.com`. Use `*` to bypass the proxy for all destinations. You must include vCenter's IP address and the IP range that you use for its machines.

additionalTrustBundle

If provided, the installation program generates a config map that is named **user-ca-bundle** in the **openshift-config** namespace to hold the additional CA certificates. If you provide **additionalTrustBundle** and at least one proxy setting, the **Proxy** object is configured to reference the **user-ca-bundle** config map in the **trustedCA** field. The Cluster Network Operator then creates a **trusted-ca-bundle** config map that merges the contents specified for the **trustedCA** parameter with the RHCOS trust bundle. The **additionalTrustBundle** field is required unless the proxy's identity certificate is signed by an authority from the RHCOS trust bundle.

additionalTrustBundlePolicy

Specifies the policy that determines the configuration of the **Proxy** object to reference the **user-ca-bundle** config map in the **trustedCA** field. The allowed values are **Proxyonly** and **Always**. Use **Proxyonly** to reference the **user-ca-bundle** config map only when **http/https** proxy is configured. Use **Always** to always reference the **user-ca-bundle** config map. The default value is **Proxyonly**. Optional parameter.



NOTE

The installation program does not support the proxy **readinessEndpoints** field.



NOTE

If the installer times out, restart and then complete the deployment by using the **wait-for** command of the installer. For example:

+

```
$ ./openshift-install wait-for install-complete --log-level debug
```

2. Save the file and reference it when installing OpenShift Container Platform.

The installation program creates a cluster-wide proxy that is named **cluster** that uses the proxy settings in the provided **install-config.yaml** file. If no proxy settings are provided, a **cluster Proxy** object is still created, but it will have a nil **spec**.



NOTE

Only the **Proxy** object named **cluster** is supported, and no additional proxies can be created.

2.5.5.3. Deploying with dual-stack networking

For dual-stack networking in OpenShift Container Platform clusters, you can configure IPv4 and IPv6 address endpoints for cluster nodes. To configure IPv4 and IPv6 address endpoints for cluster nodes, edit the **machineNetwork**, **clusterNetwork**, and **serviceNetwork** configuration settings in the **install-config.yaml** file. Each setting must have two CIDR entries each. For a cluster with the IPv4 family as the primary address family, specify the IPv4 setting first. For a cluster with the IPv6 family as the primary address family, specify the IPv6 setting first.

```
machineNetwork:
- cidr: {{ extcidrnet }}
- cidr: {{ extcidrnet6 }}
clusterNetwork:
- cidr: 10.128.0.0/14
  hostPrefix: 23
- cidr: fd02::/48
  hostPrefix: 64
serviceNetwork:
- 172.30.0.0/16
- fd03::/112
```

To provide an interface to the cluster for applications that use IPv4 and IPv6 addresses, configure IPv4 and IPv6 virtual IP (VIP) address endpoints for the Ingress VIP and API VIP services. To configure IPv4 and IPv6 address endpoints, edit the **apiVIPs** and **ingressVIPs** configuration settings in the **install-config.yaml** file. The **apiVIPs** and **ingressVIPs** configuration settings use a list format. The order of the list indicates the primary and secondary VIP address for each service.

```
platform:
vsphere:
  apiVIPs:
    - <api_ipv4>
    - <api_ipv6>
```

ingressVIPs:
 - <wildcard_ipv4>
 - <wildcard_ipv6>



NOTE

For a cluster with dual-stack networking configuration, you must assign both IPv4 and IPv6 addresses to the same interface.

2.5.5.4. Configuring regions and zones for a VMware vCenter

You can modify the default installation configuration file, so that you can deploy an OpenShift Container Platform cluster to multiple vSphere data centers.

The default **install-config.yaml** file configuration from the previous release of OpenShift Container Platform is deprecated. You can continue to use the deprecated default configuration, but the **openshift-installer** will prompt you with a warning message that indicates the use of deprecated fields in the configuration file.

Prerequisites

- You have an existing **install-config.yaml** installation configuration file.



IMPORTANT

You must specify at least one failure domain for your OpenShift Container Platform cluster, so that you can provision data center objects for your VMware vCenter server. Consider specifying multiple failure domains if you need to provision virtual machine nodes in different data centers, clusters, datastores, and other components. To enable regions and zones, you must define multiple failure domains for your OpenShift Container Platform cluster.

- You have installed the **govc** command line tool.



IMPORTANT

The example uses the **govc** command. The **govc** command is an open source command available from VMware; it is not available from Red Hat. The Red Hat support team does not maintain the **govc** command. Instructions for downloading and installing **govc** are found on the VMware documentation website.

Procedure

1. Create the **openshift-region** and **openshift-zone** vCenter tag categories by running the following commands:



IMPORTANT

If you specify different names for the **openshift-region** and **openshift-zone** vCenter tag categories, the installation of the OpenShift Container Platform cluster fails.

```
$ govc tags.category.create -d "OpenShift region" openshift-region
```

```
$ govc tags.category.create -d "OpenShift zone" openshift-zone
```

2. For each region where you want to deploy your cluster, create a region tag by running the following command:

```
$ govc tags.create -c <region_tag_category> <region_tag>
```

3. For each zone where you want to deploy your cluster, create a zone tag by running the following command:

```
$ govc tags.create -c <zone_tag_category> <zone_tag>
```

4. Attach region tags to each vCenter data center object by running the following command:

```
$ govc tags.attach -c <region_tag_category> <region_tag_1> /<data_center_1>
```

5. Attach the zone tags to each vCenter cluster object by running the following command:

```
$ govc tags.attach -c <zone_tag_category> <zone_tag_1> /<data_center_1>/host/<cluster1>
```

6. Change to the directory that contains the installation program and initialize the cluster deployment according to your chosen installation requirements.

Sample `install-config.yaml` file with multiple data centers defined in a vSphere center

```
# ...
compute:
---
vsphere:
  zones:
    - "<machine_pool_zone_1>"
    - "<machine_pool_zone_2>"
# ...
controlPlane:
# ...
vsphere:
  zones:
    - "<machine_pool_zone_1>"
    - "<machine_pool_zone_2>"
# ...
platform:
  vsphere:
    vcenters:
# ...
  datacenters:
    - <data_center_1_name>
    - <data_center_2_name>
failureDomains:
  - name: <machine_pool_zone_1>
    region: <region_tag_1>
    zone: <zone_tag_1>
```

```

server: <fully_qualified_domain_name>
topology:
  datacenter: <data_center_1>
  computeCluster: "/<data_center_1>/host/<cluster1>"
  networks:
    - <VM_Network1_name>
  datastore: "/<data_center_1>/datastore/<datastore1>"
  resourcePool: "/<data_center_1>/host/<cluster1>/Resources/<resourcePool1>"
  folder: "/<data_center_1>/vm/<folder1>"
- name: <machine_pool_zone_2>
  region: <region_tag_2>
  zone: <zone_tag_2>
  server: <fully_qualified_domain_name>
  topology:
    datacenter: <data_center_2>
    computeCluster: "/<data_center_2>/host/<cluster2>"
    networks:
      - <VM_Network2_name>
    datastore: "/<data_center_2>/datastore/<datastore2>"
    resourcePool: "/<data_center_2>/host/<cluster2>/Resources/<resourcePool2>"
    folder: "/<data_center_2>/vm/<folder2>"
# ...

```

2.5.5.5. Configuring host groups for a VMware vCenter



IMPORTANT

OpenShift zones support for vSphere host groups is a Technology Preview feature only. Technology Preview features are not supported with Red Hat production service level agreements (SLAs) and might not be functionally complete. Red Hat does not recommend using them in production. These features provide early access to upcoming product features, enabling customers to test functionality and provide feedback during the development process.

For more information about the support scope of Red Hat Technology Preview features, see [Technology Preview Features Support Scope](#).

You can modify the default installation configuration file to deploy an OpenShift Container Platform cluster on a VMware vSphere stretched cluster, where ESXi hosts are grouped into host groups by physical location.

The default **install-config.yaml** file configuration from previous releases of OpenShift Container Platform is deprecated. Though you can still use it, the OpenShift Container Platform installer will display a warning message that indicates the use of deprecated fields in the configuration file.

Prerequisites

- You have an existing **install-config.yaml** installation configuration file.
- You have arranged your ESXi hosts into host groups.
- You have granted the **Host.Inventory.EditCluster** privilege on the vSphere vCenter cluster object.

- You have downloaded and installed the **govc** command line tool. Instructions can be found on the VMware documentation website. Note that **govc** is an open-source tool that is not maintained by the Red Hat support team.
- You have enabled the **TechPreviewNoUpgrade** feature set. For more information, see "Enabling features using feature gates".



IMPORTANT

To enable host group support, you must define multiple failure domains for your OpenShift Container Platform cluster.

Procedure

1. Create the **openshift-region** and **openshift-zone** vCenter tag categories by running the following commands:



IMPORTANT

If you specify different names for the **openshift-region** and **openshift-zone** vCenter tag categories, the installation of the OpenShift Container Platform cluster fails.

```
$ govc tags.category.create -d "OpenShift region" openshift-region
```

```
$ govc tags.category.create -d "OpenShift zone" openshift-zone
```

2. Create a region tag for the vSphere cluster where you want to deploy your OpenShift Container Platform cluster by entering the following command:

```
$ govc tags.create -c <region_tag_category> <region_tag>
```

3. Create a zone tag for each host group by entering the following command as needed:

```
$ govc tags.create -c <zone_tag_category> <zone_tag>
```

4. Attach the region tag to the vCenter cluster object by entering the following command:

```
$ govc tags.attach -c <region_tag_category> <region_tag_1>
/<datacenter_1>/host/<cluster_1>
```

5. Use zone tags to associate each ESXi host with its host group, by entering the following command for each ESXi host:

```
$ govc tags.attach -c <zone_tag_category> <zone_tag_for_host_group_1>
/<datacenter_1>/host/<cluster_1>/<esxi_host_in_host_group_1>
```

6. Change to the directory that contains the installation program and initialize the cluster deployment according to your chosen installation requirements.

Sample `install-config.yaml` file with multiple host groups



```

featureSet: TechPreviewNoUpgrade
featureGate:
  - "VSphereHostVMGroupZonal=true"
# ...
platform:
  vsphere:
    vcenters:
# ...
  datacenters:
    - <data_center_1_name>
failureDomains:
  - name: <host_group_1>
    region: <cluster_1_region_tag>
    zone: <host_group_1_zone_tag>
    regionType: "ComputeCluster"
    zoneType: "HostGroup"
    server: <fully_qualified_domain_name>
    topology:
      datacenter: <data_center_1>
      computeCluster: "/<data_center_1>/host/<cluster_1>"
      networks:
        - <VM_Network1_name>
        hostGroup: <host_group_1_name>
        datastore: "/<data_center_1>/datastore/<datastore_1>"
        resourcePool: "/<data_center_1>/host/<cluster_1>/Resources/<resourcePool_1>"
        folder: "/<data_center_1>/vm/<folder_1>"
  - name: <host_group_2>
    region: <cluster_1_region_tag>
    zone: <host_group_2_zone_tag>
    regionType: "ComputeCluster"
    zoneType: "HostGroup"
    server: <fully_qualified_domain_name>
    topology:
      datacenter: <data_center_1>
      computeCluster: "/<data_center_1>/host/<cluster_1>"
      networks:
        - <VM_Network1_name>
        hostGroup: <host_group_2_name>
        datastore: "/<data_center_1>/datastore/<datastore_1>"
        resourcePool: "/<data_center_1>/host/<cluster_1>/Resources/<resourcePool_1>"
        folder: "/<data_center_1>/vm/<folder_1>"
```

Additional resources

- Enabling features using feature gates

2.5.5.6. Configuring multiple NICs

For scenarios requiring multiple network interface controller (NIC), you can configure multiple network adapters per node.



IMPORTANT

Configuring multiple NICs is a Technology Preview feature only. Technology Preview features are not supported with Red Hat production service level agreements (SLAs) and might not be functionally complete. Red Hat does not recommend using them in production. These features provide early access to upcoming product features, enabling customers to test functionality and provide feedback during the development process.

For more information about the support scope of Red Hat Technology Preview features, see [Technology Preview Features Support Scope](#).

Procedure

- Specify the network adapter names in the networks section of **platform.vsphere.failureDomains[*].topology** as shown in the following **install-config.yaml** file:

```
platform:
  vsphere:
    vcenters:
      ...
    failureDomains:
      - name: <failure_domain_name>
        region: <default_region_name>
        zone: <default_zone_name>
        server: <fully_qualified_domain_name>
        topology:
          datacenter: <data_center>
          computeCluster: "/<data_center>/host/<cluster>"
        networks: ①
          - <VM_network1_name>
          - <VM_network2_name>
          - ...
          - <VM_network10_name>
```

① Specifies the list of network adapters. You can specify up to 10 network adapters.

- Specify at least one of the following configurations in the **install-config.yaml** file:

- **networking.machineNetwork**

Example configuration

```
networking:
  ...
  machineNetwork:
    - cidr: 10.0.0.0/16
  ...
```



NOTE

The **networking.machineNetwork.cidr** field must correspond to an address on the first adapter defined in **topology.networks**.

- Add a **nodeNetworking** object to the **install-config.yaml** file and specify internal and external network subnet CIDR implementations for the object.

Example configuration

```
platform:
vsphere:
  nodeNetworking:
    external:
      networkSubnetCidr:
        - <machine_network_cidr_ipv4>
        - <machine_network_cidr_ipv6>
    internal:
      networkSubnetCidr:
        - <machine_network_cidr_ipv4>
        - <machine_network_cidr_ipv6>
  failureDomains:
    - name: <failure_domain_name>
      region: <default_region_name>
```

Additional resources

- [Network configuration parameters](#)

2.5.6. Network configuration phases

There are two phases prior to OpenShift Container Platform installation where you can customize the network configuration.

Phase 1

You can customize the following network-related fields in the **install-config.yaml** file before you create the manifest files:

- **networking.networkType**
- **networking.clusterNetwork**
- **networking.serviceNetwork**
- **networking.machineNetwork**
- **nodeNetworking**

For more information, see "Installation configuration parameters".



NOTE

Set the **networking.machineNetwork** to match the Classless Inter-Domain Routing (CIDR) where the preferred subnet is located.



IMPORTANT

The CIDR range **172.17.0.0/16** is reserved by **libVirt**. You cannot use any other CIDR range that overlaps with the **172.17.0.0/16** CIDR range for networks in your cluster.

Phase 2

After creating the manifest files by running **openshift-install create manifests**, you can define a customized Cluster Network Operator manifest with only the fields you want to modify. You can use the manifest to specify an advanced network configuration.

During phase 2, you cannot override the values that you specified in phase 1 in the **install-config.yaml** file. However, you can customize the network plugin during phase 2.

2.5.7. Specifying advanced network configuration

You can use advanced network configuration for your network plugin to integrate your cluster into your existing network environment.

You can specify advanced network configuration only before you install the cluster.



IMPORTANT

Customizing your network configuration by modifying the OpenShift Container Platform manifest files created by the installation program is not supported. Applying a manifest file that you create, as in the following procedure, is supported.

Prerequisites

- You have created the **install-config.yaml** file and completed any modifications to it.

Procedure

1. Change to the directory that contains the installation program and create the manifests:

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

1 `<installation_directory>` specifies the name of the directory that contains the **install-config.yaml** file for your cluster.

2. Create a stub manifest file for the advanced network configuration that is named **cluster-network-03-config.yaml** in the `<installation_directory>/manifests` directory:

```
apiVersion: operator.openshift.io/v1
kind: Network
metadata:
  name: cluster
spec:
```

3. Specify the advanced network configuration for your cluster in the **cluster-network-03-config.yaml** file, such as in the following example:

Enable IPsec for the OVN-Kubernetes network provider

```
apiVersion: operator.openshift.io/v1
kind: Network
metadata:
  name: cluster
```

```

spec:
  defaultNetwork:
    ovnKubernetesConfig:
      ipsecConfig:
        mode: Full

```

4. Optional: Back up the **manifests/cluster-network-03-config.yml** file. The installation program consumes the **manifests**/ directory when you create the Ignition config files.

2.5.7.1. Specifying multiple subnets for your network

Before you install an OpenShift Container Platform cluster on a vSphere host, you can specify multiple subnets for a networking implementation so that the vSphere cloud controller manager (CCM) can select the appropriate subnet for a given networking situation. vSphere can use the subnet for managing pods and services on your cluster.

For this configuration, you must specify internal and external Classless Inter-Domain Routing (CIDR) implementations in the vSphere CCM configuration. Each CIDR implementation lists an IP address range that the CCM uses to decide what subnets interact with traffic from internal and external networks.



IMPORTANT

Failure to configure internal and external CIDR implementations in the vSphere CCM configuration can cause the vSphere CCM to select the wrong subnet. This situation causes the following error:

```

ERROR Bootstrap failed to complete: timed out waiting for the condition
ERROR Failed to wait for bootstrapping to complete. This error usually happens when
there is a problem with control plane hosts that prevents the control plane operators
from creating the control plane.

```

This configuration can cause new nodes that associate with a **MachineSet** object with a single subnet to become unusable as each new node receives the **node.cloudprovider.kubernetes.io/uninitialized** taint. These situations can cause communication issues with the Kubernetes API server that can cause installation of the cluster to fail.

Prerequisites

- You created Kubernetes manifest files for your OpenShift Container Platform cluster.

Procedure

1. From the directory where you store your OpenShift Container Platform cluster manifest files, open the **manifests/cluster-infrastructure-02-config.yml** manifest file.
2. Add a **nodeNetworking** object to the file and specify internal and external network subnet CIDR implementations for the object.

TIP

For most networking situations, consider setting the standard multiple-subnet configuration. This configuration requires that you set the same IP address ranges in the **nodeNetworking.internal.networkSubnetCidr** and **nodeNetworking.external.networkSubnetCidr** parameters.

Example of a configured `cluster-infrastructure-02-config.yml` manifest file

```
apiVersion: config.openshift.io/v1
kind: Infrastructure
metadata:
  name: cluster
spec:
  cloudConfig:
    key: config
    name: cloud-provider-config
  platformSpec:
    type: VSphere
    vsphere:
      failureDomains:
        - name: generated-failure-domain
      ...
      nodeNetworking:
        external:
          networkSubnetCidr:
            - <machine_network_cidr_ipv4>
            - <machine_network_cidr_ipv6>
        internal:
          networkSubnetCidr:
            - <machine_network_cidr_ipv4>
            - <machine_network_cidr_ipv6>
  # ...
```

Additional resources

- [Cluster Network Operator configuration](#)
- [.spec.platformSpec.vsphere.nodeNetworking](#)

2.5.8. Cluster Network Operator configuration

To manage cluster networking, configure the Cluster Network Operator (CNO) **Network** custom resource (CR) named **cluster** so the cluster uses the correct IP ranges and network plugin settings for reliable pod and service connectivity. Some settings and fields are inherited at the time of install or by the **default.Network.type** plugin, OVN-Kubernetes.

The CNO configuration inherits the following fields during cluster installation from the **Network** API in the **Network.config.openshift.io** API group:

clusterNetwork

IP address pools from which pod IP addresses are allocated.

serviceNetwork

IP address pool for services.

defaultNetwork.type

Cluster network plugin. **OVNKubernetes** is the only supported plugin during installation.

You can specify the cluster network plugin configuration for your cluster by setting the fields for the **defaultNetwork** object in the CNO object named **cluster**.

2.5.8.1. Cluster Network Operator configuration object

The fields for the Cluster Network Operator (CNO) are described in the following table:

Table 2.7. Cluster Network Operator configuration object

Field	Type	Description
metadata.name	string	The name of the CNO object. This name is always cluster .
spec.clusterNetwork	array	<p>A list specifying the blocks of IP addresses from which pod IP addresses are allocated and the subnet prefix length assigned to each individual node in the cluster. For example:</p> <pre>spec: clusterNetwork: - cidr: 10.128.0.0/19 hostPrefix: 23 - cidr: 10.128.32.0/19 hostPrefix: 23</pre>
spec.serviceNetwork	array	<p>A block of IP addresses for services. The OVN-Kubernetes network plugin supports only a single IP address block for the service network. For example:</p> <pre>spec: serviceNetwork: - 172.30.0.0/14</pre> <p>You can customize this field only in the install-config.yaml file before you create the manifests. The value is read-only in the manifest file.</p>
spec.defaultNetwork	object	Configures the network plugin for the cluster network.

Field	Type	Description
spec.additionalRoutingCapabilities.providers	array	<p>This setting enables a dynamic routing provider. The FRR routing capability provider is required for the route advertisement feature. The only supported value is FRR.</p> <ul style="list-style-type: none"> • FRR: The FRR routing provider <pre>spec: additionalRoutingCapabilities: providers: - FRR</pre>



IMPORTANT

For a cluster that needs to deploy objects across multiple networks, ensure that you specify the same value for the **clusterNetwork.hostPrefix** parameter for each network type that is defined in the **install-config.yaml** file. Setting a different value for each **clusterNetwork.hostPrefix** parameter can impact the OVN-Kubernetes network plugin, where the plugin cannot effectively route object traffic among different nodes.

2.5.8.2. **defaultNetwork** object configuration

The values for the **defaultNetwork** object are defined in the following table:

Table 2.8. **defaultNetwork** object

Field	Type	Description
type	string	<p>OVNKubernetes. The Red Hat OpenShift Networking network plugin is selected during installation. This value cannot be changed after cluster installation.</p> <div style="display: flex; align-items: center;">  <div style="margin-left: 10px;"> NOTE OpenShift Container Platform uses the OVN-Kubernetes network plugin by default. </div> </div>
ovnKubernetesConfig	object	<p>This object is only valid for the OVN-Kubernetes network plugin.</p>

2.5.8.3. Configuration for the OVN-Kubernetes network plugin

The following table describes the configuration fields for the OVN-Kubernetes network plugin:

Table 2.9. **ovnKubernetesConfig** object

Field	Type	Description
mtu	integer	<p>The maximum transmission unit (MTU) for the Geneve (Generic Network Virtualization Encapsulation) overlay network. This is detected automatically based on the MTU of the primary network interface. You do not normally need to override the detected MTU.</p> <p>If the auto-detected value is not what you expect it to be, confirm that the MTU on the primary network interface on your nodes is correct. You cannot use this option to change the MTU value of the primary network interface on the nodes.</p> <p>If your cluster requires different MTU values for different nodes, you must set this value to 100 less than the lowest MTU value in your cluster. For example, if some nodes in your cluster have an MTU of 9001, and some have an MTU of 1500, you must set this value to 1400.</p>
genevePort	integer	The port to use for all Geneve packets. The default value is 6081 . This value cannot be changed after cluster installation.
ipsecConfig	object	Specify a configuration object for customizing the IPsec configuration.
ipv4	object	Specifies a configuration object for IPv4 settings.
ipv6	object	Specifies a configuration object for IPv6 settings.
policyAuditConfig	object	Specify a configuration object for customizing network policy audit logging. If unset, the defaults audit log settings are used.
routeAdvertises ments	string	<p>Specifies whether to advertise cluster network routes. The default value is Disabled.</p> <ul style="list-style-type: none"> ● Enabled: Import routes to the cluster network and advertise cluster network routes as configured in RouteAdvertisements objects. ● Disabled: Do not import routes to the cluster network or advertise cluster network routes.

Field	Type	Description
gatewayConfig	object	<p>Optional: Specify a configuration object for customizing how egress traffic is sent to the node gateway. Valid values are Shared and Local. The default value is Shared. In the default setting, the Open vSwitch (OVS) outputs traffic directly to the node IP interface. In the Local setting, it traverses the host network; consequently, it gets applied to the routing table of the host.</p> <p> NOTE</p> <p>While migrating egress traffic, you can expect some disruption to workloads and service traffic until the Cluster Network Operator (CNO) successfully rolls out the changes.</p>

Table 2.10. `ovnKubernetesConfig.ipv4` object

Field	Type	Description
internalTransitSwitchSubnet	string	<p>If your existing network infrastructure overlaps with the 100.88.0.0/16 IPv4 subnet, you can specify a different IP address range for internal use by OVN-Kubernetes. The subnet for the distributed transit switch that enables east-west traffic. This subnet cannot overlap with any other subnets used by OVN-Kubernetes or on the host itself. It must be large enough to accommodate one IP address per node in your cluster.</p> <p>The default value is 100.88.0.0/16.</p>
internalJoinSubnet	string	<p>If your existing network infrastructure overlaps with the 100.64.0.0/16 IPv4 subnet, you can specify a different IP address range for internal use by OVN-Kubernetes. You must ensure that the IP address range does not overlap with any other subnet used by your OpenShift Container Platform installation. The IP address range must be larger than the maximum number of nodes that can be added to the cluster. For example, if the clusterNetwork.cidr value is 10.128.0.0/14 and the clusterNetwork.hostPrefix value is /23, then the maximum number of nodes is 2^(23-14)=512.</p> <p>The default value is 100.64.0.0/16.</p>

Table 2.11. `ovnKubernetesConfig.ipv6` object

Field	Type	Description
internalTransitSwitchSubnet	string	<p>If your existing network infrastructure overlaps with the fd97::/64 IPv6 subnet, you can specify a different IP address range for internal use by OVN-Kubernetes. The subnet for the distributed transit switch that enables east-west traffic. This subnet cannot overlap with any other subnets used by OVN-Kubernetes or on the host itself. It must be large enough to accommodate one IP address per node in your cluster.</p> <p>The default value is fd97::/64.</p>
internalJoinSubnet	string	<p>If your existing network infrastructure overlaps with the fd98::/64 IPv6 subnet, you can specify a different IP address range for internal use by OVN-Kubernetes. You must ensure that the IP address range does not overlap with any other subnet used by your OpenShift Container Platform installation. The IP address range must be larger than the maximum number of nodes that can be added to the cluster.</p> <p>The default value is fd98::/64.</p>

Table 2.12. **policyAuditConfig** object

Field	Type	Description
rateLimit	integer	The maximum number of messages to generate every second per node. The default value is 20 messages per second.
maxFileSize	integer	The maximum size for the audit log in bytes. The default value is 50000000 or 50 MB.
maxLogFiles	integer	The maximum number of log files that are retained.
destination	string	<p>One of the following additional audit log targets:</p> <p>libc The libc syslog() function of the journald process on the host.</p> <p>udp:<host>:<port> A syslog server. Replace <host>:<port> with the host and port of the syslog server.</p> <p>unix:<file> A Unix Domain Socket file specified by <file>.</p> <p>null Do not send the audit logs to any additional target.</p>
syslogFacility	string	The syslog facility, such as kern , as defined by RFC5424. The default value is local0 .

Table 2.13. `gatewayConfig` object

Field	Type	Description
<code>routingViaHost</code>	<code>boolean</code>	<p>Set this field to <code>true</code> to send egress traffic from pods to the host networking stack. For highly-specialized installations and applications that rely on manually configured routes in the kernel routing table, you might want to route egress traffic to the host networking stack. By default, egress traffic is processed in OVN to exit the cluster and is not affected by specialized routes in the kernel routing table. The default value is <code>false</code>.</p> <p>This field has an interaction with the Open vSwitch hardware offloading feature. If you set this field to <code>true</code>, you do not receive the performance benefits of the offloading because egress traffic is processed by the host networking stack.</p>
<code>ipForwarding</code>	<code>object</code>	<p>You can control IP forwarding for all traffic on OVN-Kubernetes managed interfaces by using the <code>ipForwarding</code> specification in the <code>Network</code> resource. Specify <code>Restricted</code> to only allow IP forwarding for Kubernetes related traffic. Specify <code>Global</code> to allow forwarding of all IP traffic. For new installations, the default is <code>Restricted</code>. For updates to OpenShift Container Platform 4.14 or later, the default is <code>Global</code>.</p> <div style="display: flex; align-items: center;">  <div style="margin-left: 10px;"> <p>NOTE</p> <p>The default value of <code>Restricted</code> sets the IP forwarding to drop.</p> </div> </div>
<code>ipv4</code>	<code>object</code>	Optional: Specify an object to configure the internal OVN-Kubernetes masquerade address for host to service traffic for IPv4 addresses.
<code>ipv6</code>	<code>object</code>	Optional: Specify an object to configure the internal OVN-Kubernetes masquerade address for host to service traffic for IPv6 addresses.

Table 2.14. `gatewayConfig.ipv4` object

Field	Type	Description
-------	------	-------------

Field	Type	Description
internalMasqueradeSubnet	string	<p>The masquerade IPv4 addresses that are used internally to enable host to service traffic. The host is configured with these IP addresses as well as the shared gateway bridge interface. The default value is 169.254.169.0/29.</p> <p> IMPORTANT</p> <p>For OpenShift Container Platform 4.17 and later versions, clusters use 169.254.0.0/17 as the default masquerade subnet. For upgraded clusters, there is no change to the default masquerade subnet.</p>

Table 2.15. `gatewayConfig.ipv6` object

Field	Type	Description
internalMasqueradeSubnet	string	<p>The masquerade IPv6 addresses that are used internally to enable host to service traffic. The host is configured with these IP addresses as well as the shared gateway bridge interface. The default value is fd69::/125.</p> <p> IMPORTANT</p> <p>For OpenShift Container Platform 4.17 and later versions, clusters use fd69::/112 as the default masquerade subnet. For upgraded clusters, there is no change to the default masquerade subnet.</p>

Table 2.16. `ipsecConfig` object

Field	Type	Description
mode	string	<p>Specifies the behavior of the IPsec implementation. Must be one of the following values:</p> <ul style="list-style-type: none"> • Disabled: IPsec is not enabled on cluster nodes. • External: IPsec is enabled for network traffic with external hosts. • Full: IPsec is enabled for pod traffic and network traffic with external hosts.

Example OVN-Kubernetes configuration with IPsec enabled

```
defaultNetwork:  
  type: OVNKubernetes  
  ovnKubernetesConfig:  
    mtu: 1400  
    genevePort: 6081  
    ipsecConfig:  
      mode: Full
```

2.5.9. Services for a user-managed load balancer

To integrate your infrastructure with existing network standards or gain more control over traffic management in OpenShift Container Platform, configure services for a user-managed load balancer.



IMPORTANT

Configuring a user-managed load balancer depends on your vendor's load balancer.

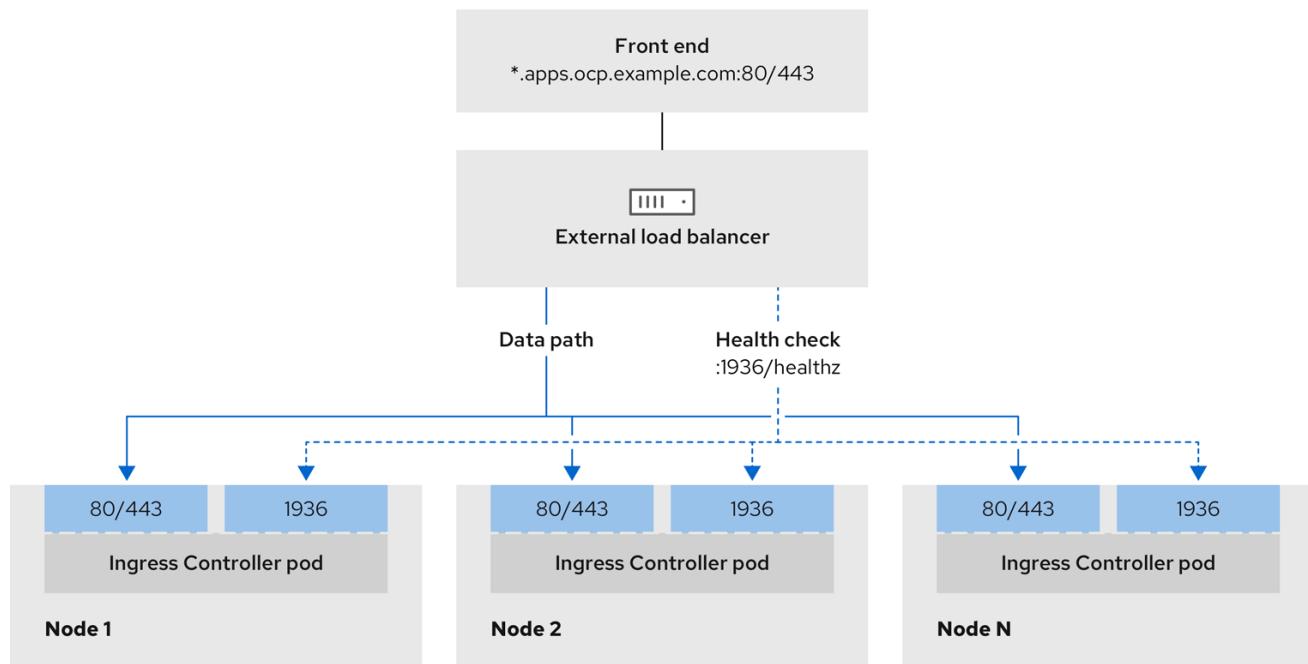
The information and examples in this section are for guideline purposes only. Consult the vendor documentation for more specific information about the vendor's load balancer.

Red Hat supports the following services for a user-managed load balancer:

- Ingress Controller
- OpenShift API
- OpenShift MachineConfig API

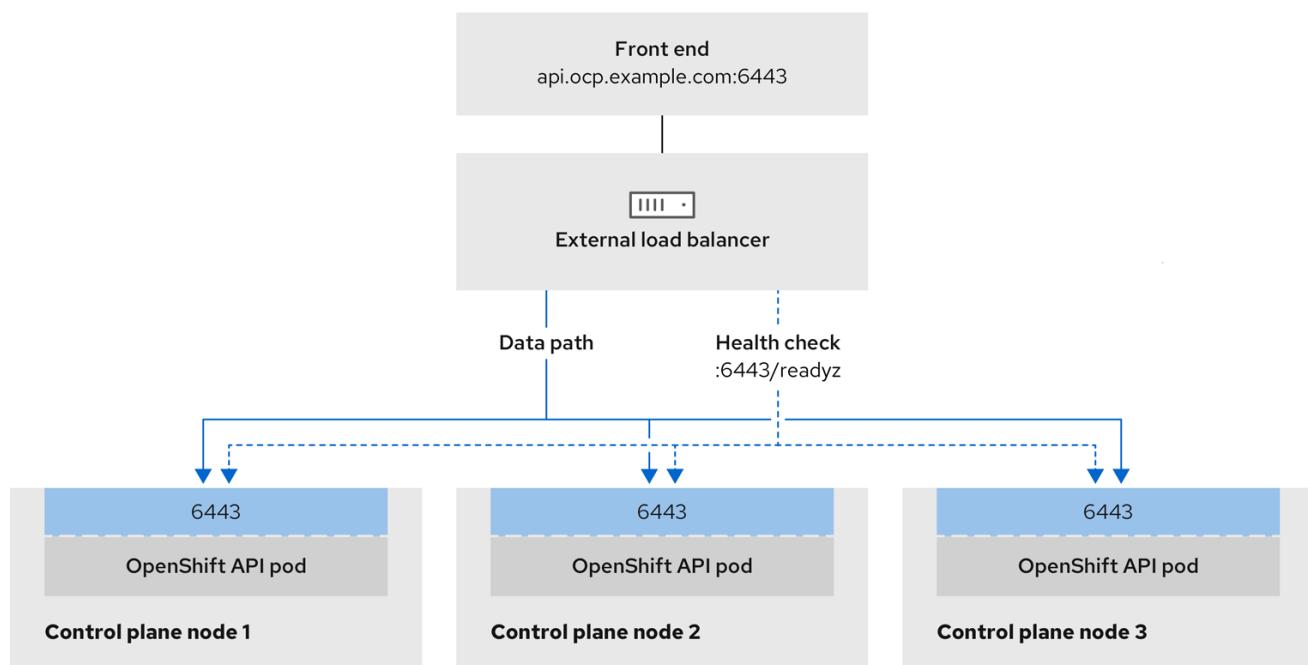
You can choose whether you want to configure one or all of these services for a user-managed load balancer. Configuring only the Ingress Controller service is a common configuration option. To better understand each service, view the following diagrams:

Figure 2.4. Example network workflow that shows an Ingress Controller operating in an OpenShift Container Platform environment



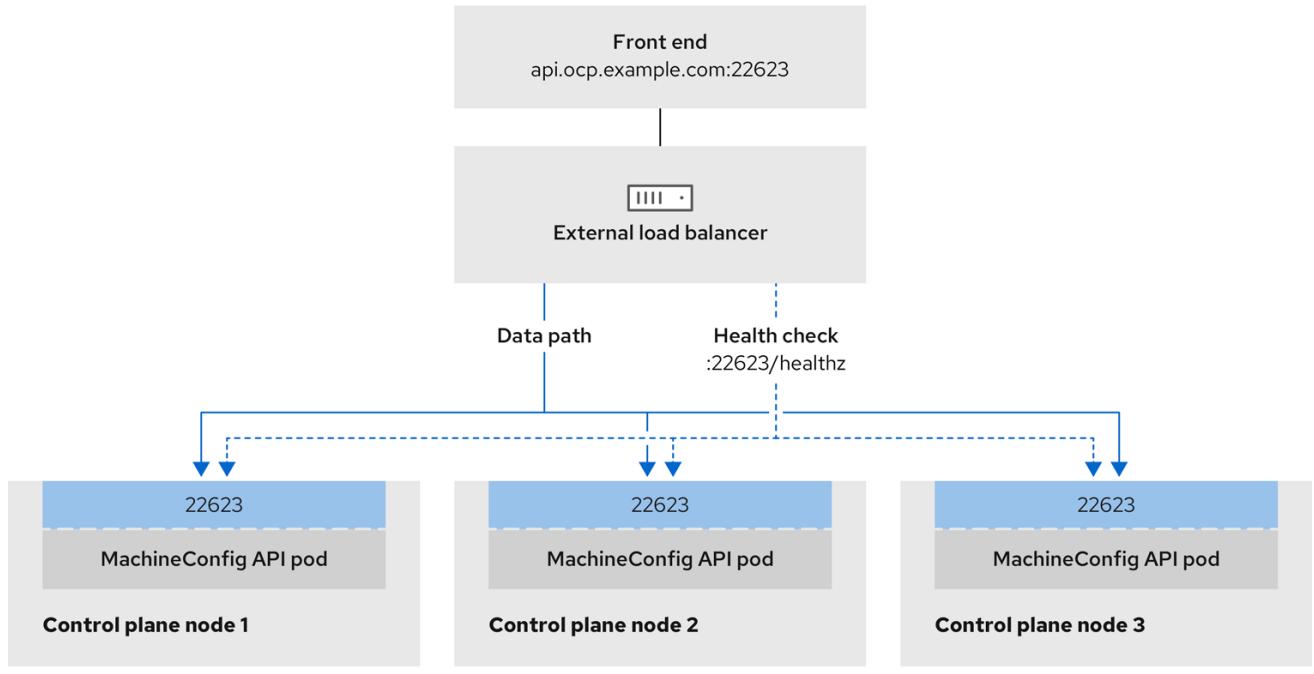
496_OpenShift_I223

Figure 2.5. Example network workflow that shows an OpenShift API operating in an OpenShift Container Platform environment



496_OpenShift_I223

Figure 2.6. Example network workflow that shows an OpenShift MachineConfig API operating in an OpenShift Container Platform environment



496_OpenShift_I223

The following configuration options are supported for user-managed load balancers:

- Use a node selector to map the Ingress Controller to a specific set of nodes. You must assign a static IP address to each node in this set, or configure each node to receive the same IP address from the Dynamic Host Configuration Protocol (DHCP). Infrastructure nodes commonly receive this type of configuration.
- Target all IP addresses on a subnet. This configuration can reduce maintenance overhead, because you can create and destroy nodes within those networks without reconfiguring the load balancer targets. If you deploy your ingress pods by using a machine set on a smaller network, such as a /27 or /28, you can simplify your load balancer targets.

TIP

You can list all IP addresses that exist in a network by checking the machine config pool's resources.

Before you configure a user-managed load balancer for your OpenShift Container Platform cluster, consider the following information:

- For a front-end IP address, you can use the same IP address for the front-end IP address, the Ingress Controller's load balancer, and API load balancer. Check the vendor's documentation for this capability.
- For a back-end IP address, ensure that an IP address for an OpenShift Container Platform control plane node does not change during the lifetime of the user-managed load balancer. You can achieve this by completing one of the following actions:
 - Assign a static IP address to each control plane node.

- Configure each node to receive the same IP address from the DHCP every time the node requests a DHCP lease. Depending on the vendor, the DHCP lease might be in the form of an IP reservation or a static DHCP assignment.
- Manually define each node that runs the Ingress Controller in the user-managed load balancer for the Ingress Controller back-end service. For example, if the Ingress Controller moves to an undefined node, a connection outage can occur.

2.5.9.1. Configuring a user-managed load balancer

To integrate your infrastructure with existing network standards or gain more control over traffic management in OpenShift Container Platform, use a user-managed load balancer in place of the default load balancer.



IMPORTANT

Before you configure a user-managed load balancer, ensure that you read the "Services for a user-managed load balancer" section.

Read the following prerequisites that apply to the service that you want to configure for your user-managed load balancer.



NOTE

MetalLB, which runs on a cluster, functions as a user-managed load balancer.

Prerequisites

The following list details OpenShift API prerequisites:

- You defined a front-end IP address.
- TCP ports 6443 and 22623 are exposed on the front-end IP address of your load balancer. Check the following items:
 - Port 6443 provides access to the OpenShift API service.
 - Port 22623 can provide ignition startup configurations to nodes.
- The front-end IP address and port 6443 are reachable by all users of your system with a location external to your OpenShift Container Platform cluster.
- The front-end IP address and port 22623 are reachable only by OpenShift Container Platform nodes.
- The load balancer backend can communicate with OpenShift Container Platform control plane nodes on port 6443 and 22623.

The following list details Ingress Controller prerequisites:

- You defined a front-end IP address.
- TCP port 443 and port 80 are exposed on the front-end IP address of your load balancer.
- The front-end IP address, port 80 and port 443 are reachable by all users of your system with a location external to your OpenShift Container Platform cluster.

- The front-end IP address, port 80 and port 443 are reachable by all nodes that operate in your OpenShift Container Platform cluster.
- The load balancer backend can communicate with OpenShift Container Platform nodes that run the Ingress Controller on ports 80, 443, and 1936.

The following list details prerequisites for health check URL specifications:

You can configure most load balancers by setting health check URLs that determine if a service is available or unavailable. OpenShift Container Platform provides these health checks for the OpenShift API, Machine Configuration API, and Ingress Controller backend services.

The following example shows a Kubernetes API health check specification for a backend service:

```
Path: HTTPS:6443/readyz
Healthy threshold: 2
Unhealthy threshold: 2
Timeout: 10
Interval: 10
```

The following example shows a Machine Config API health check specification for a backend service:

```
Path: HTTPS:22623/healthz
Healthy threshold: 2
Unhealthy threshold: 2
Timeout: 10
Interval: 10
```

The following example shows a Ingress Controller health check specification for a backend service:

```
Path: HTTP:1936/healthz/ready
Healthy threshold: 2
Unhealthy threshold: 2
Timeout: 5
Interval: 10
```

Procedure

1. Configure the HAProxy Ingress Controller, so that you can enable access to the cluster from your load balancer on ports 6443, 22623, 443, and 80. Depending on your needs, you can specify the IP address of a single subnet or IP addresses from multiple subnets in your HAProxy configuration.

Example HAProxy configuration with one listed subnet

```
# ...
listen my-cluster-api-6443
  bind 192.168.1.100:6443
  mode tcp
  balance roundrobin
  option httpchk
  http-check connect
  http-check send meth GET uri /readyz
  http-check expect status 200
  server my-cluster-master-2 192.168.1.101:6443 check inter 10s rise 2 fall 2
```

```

server my-cluster-master-0 192.168.1.102:6443 check inter 10s rise 2 fall 2
server my-cluster-master-1 192.168.1.103:6443 check inter 10s rise 2 fall 2

listen my-cluster-machine-config-api-22623
  bind 192.168.1.100:22623
  mode tcp
  balance roundrobin
  option httpchk
  http-check connect
  http-check send meth GET uri /healthz
  http-check expect status 200
    server my-cluster-master-2 192.168.1.101:22623 check inter 10s rise 2 fall 2
    server my-cluster-master-0 192.168.1.102:22623 check inter 10s rise 2 fall 2
    server my-cluster-master-1 192.168.1.103:22623 check inter 10s rise 2 fall 2

listen my-cluster-apps-443
  bind 192.168.1.100:443
  mode tcp
  balance roundrobin
  option httpchk
  http-check connect
  http-check send meth GET uri /healthz/ready
  http-check expect status 200
    server my-cluster-worker-0 192.168.1.111:443 check port 1936 inter 10s rise 2 fall 2
    server my-cluster-worker-1 192.168.1.112:443 check port 1936 inter 10s rise 2 fall 2
    server my-cluster-worker-2 192.168.1.113:443 check port 1936 inter 10s rise 2 fall 2

listen my-cluster-apps-80
  bind 192.168.1.100:80
  mode tcp
  balance roundrobin
  option httpchk
  http-check connect
  http-check send meth GET uri /healthz/ready
  http-check expect status 200
    server my-cluster-worker-0 192.168.1.111:80 check port 1936 inter 10s rise 2 fall 2
    server my-cluster-worker-1 192.168.1.112:80 check port 1936 inter 10s rise 2 fall 2
    server my-cluster-worker-2 192.168.1.113:80 check port 1936 inter 10s rise 2 fall 2
# ...

```

Example HAProxy configuration with multiple listed subnets

```

# ...
listen api-server-6443
  bind *:6443
  mode tcp
    server master-00 192.168.83.89:6443 check inter 1s
    server master-01 192.168.84.90:6443 check inter 1s
    server master-02 192.168.85.99:6443 check inter 1s
    server bootstrap 192.168.80.89:6443 check inter 1s

listen machine-config-server-22623
  bind *:22623
  mode tcp
    server master-00 192.168.83.89:22623 check inter 1s
    server master-01 192.168.84.90:22623 check inter 1s

```

```

server master-02 192.168.85.99:22623 check inter 1s
server bootstrap 192.168.80.89:22623 check inter 1s

listen ingress-router-80
  bind *:80
  mode tcp
  balance source
    server worker-00 192.168.83.100:80 check inter 1s
    server worker-01 192.168.83.101:80 check inter 1s

listen ingress-router-443
  bind *:443
  mode tcp
  balance source
    server worker-00 192.168.83.100:443 check inter 1s
    server worker-01 192.168.83.101:443 check inter 1s

listen ironic-api-6385
  bind *:6385
  mode tcp
  balance source
    server master-00 192.168.83.89:6385 check inter 1s
    server master-01 192.168.84.90:6385 check inter 1s
    server master-02 192.168.85.99:6385 check inter 1s
    server bootstrap 192.168.80.89:6385 check inter 1s

listen inspector-api-5050
  bind *:5050
  mode tcp
  balance source
    server master-00 192.168.83.89:5050 check inter 1s
    server master-01 192.168.84.90:5050 check inter 1s
    server master-02 192.168.85.99:5050 check inter 1s
    server bootstrap 192.168.80.89:5050 check inter 1s
# ...

```

2. Use the **curl** CLI command to verify that the user-managed load balancer and its resources are operational:

a. Verify that the cluster machine configuration API is accessible to the Kubernetes API server resource, by running the following command and observing the response:

```
$ curl https://<loadbalancer_ip_address>:6443/version --insecure
```

If the configuration is correct, you receive a JSON object in response:

```
{
  "major": "1",
  "minor": "11+",
  "gitVersion": "v1.11.0+ad103ed",
  "gitCommit": "ad103ed",
  "gitTreeState": "clean",
  "buildDate": "2019-01-09T06:44:10Z",
  "goVersion": "go1.10.3",
  "compiler": "gc",
  "platform": "linux/amd64"
```

```
    }
```

b. Verify that the cluster machine configuration API is accessible to the Machine config server resource, by running the following command and observing the output:

```
$ curl -v https://<loadbalancer_ip_address>:22623/healthz --insecure
```

If the configuration is correct, the output from the command shows the following response:

```
HTTP/1.1 200 OK
Content-Length: 0
```

c. Verify that the controller is accessible to the Ingress Controller resource on port 80, by running the following command and observing the output:

```
$ curl -I -L -H "Host: console-openshift-console.apps.<cluster_name>.<base_domain>"  
http://<load_balancer_front_end_IP_address>
```

If the configuration is correct, the output from the command shows the following response:

```
HTTP/1.1 302 Found
content-length: 0
location: https://console-openshift-console.apps.ocp4.private.opequon.net/
cache-control: no-cache
```

d. Verify that the controller is accessible to the Ingress Controller resource on port 443, by running the following command and observing the output:

```
$ curl -I -L --insecure --resolve console-openshift-console.apps.<cluster_name>.  
<base_domain>:443:<Load Balancer Front End IP Address> https://console-openshift-  
console.apps.<cluster_name>.<base_domain>
```

If the configuration is correct, the output from the command shows the following response:

```
HTTP/1.1 200 OK
referrer-policy: strict-origin-when-cross-origin
set-cookie: csrf-
token=UIYWOyQ62LWjw2h003xtYSKlh1a0Py2hhctw0WmV2YEdhJjFyQwWcGBsja261dG
LgaYO0nxzVErhiXt6QepA7g==; Path=/; Secure; SameSite=Lax
x-content-type-options: nosniff
x-dns-prefetch-control: off
x-frame-options: DENY
x-xss-protection: 1; mode=block
date: Wed, 04 Oct 2023 16:29:38 GMT
content-type: text/html; charset=utf-8
set-cookie:
1e2670d92730b515ce3a1bb65da45062=1bf5e9573c9a2760c964ed1659cc1673; path=/;
HttpOnly; Secure; SameSite=None
cache-control: private
```

3. Configure the DNS records for your cluster to target the front-end IP addresses of the user-managed load balancer. You must update records to your DNS server for the cluster API and applications over the load balancer. The following examples shows modified DNS records:

—

<load_balancer_ip_address> **A** api.<cluster_name>.<base_domain>
A record pointing to Load Balancer Front End

<load_balancer_ip_address> **A** apps.<cluster_name>.<base_domain>
A record pointing to Load Balancer Front End



IMPORTANT

DNS propagation might take some time for each DNS record to become available. Ensure that each DNS record propagates before validating each record.

4. For your OpenShift Container Platform cluster to use the user-managed load balancer, you must specify the following configuration in your cluster's **install-config.yaml** file:

```
# ...
platform:
  vsphere:
    loadBalancer:
      type: UserManaged
    apiVIPs:
      - <api_ip> ①
    ingressVIPs:
      - <ingress_ip> ②
# ...
```

where:

loadBalancer.type

Set **UserManaged** for the **type** parameter to specify a user-managed load balancer for your cluster. The parameter defaults to **OpenShiftManagedDefault**, which denotes the default internal load balancer. For services defined in an **openshift-kni-infra** namespace, a user-managed load balancer can deploy the **coredns** service to pods in your cluster but ignores **keepalived** and **haproxy** services.

loadBalancer.<api_ip>

Specifies a user-managed load balancer. Specify the user-managed load balancer's public IP address, so that the Kubernetes API can communicate with the user-managed load balancer. Mandatory parameter.

loadBalancer.<ingress_ip>

Specifies a user-managed load balancer. Specify the user-managed load balancer's public IP address, so that the user-managed load balancer can manage ingress traffic for your cluster. Mandatory parameter.

Verification

1. Use the **curl** CLI command to verify that the user-managed load balancer and DNS record configuration are operational:
 - a. Verify that you can access the cluster API, by running the following command and observing the output:

```
$ curl https://api.<cluster_name>.<base_domain>:6443/version --insecure
```

If the configuration is correct, you receive a JSON object in response:

```
{
  "major": "1",
  "minor": "11+",
  "gitVersion": "v1.11.0+ad103ed",
  "gitCommit": "ad103ed",
  "gitTreeState": "clean",
  "buildDate": "2019-01-09T06:44:10Z",
  "goVersion": "go1.10.3",
  "compiler": "gc",
  "platform": "linux/amd64"
}
```

- b. Verify that you can access the cluster machine configuration, by running the following command and observing the output:

```
$ curl -v https://api.<cluster_name>.<base_domain>:22623/healthz --insecure
```

If the configuration is correct, the output from the command shows the following response:

```
HTTP/1.1 200 OK
Content-Length: 0
```

- c. Verify that you can access each cluster application on port 80, by running the following command and observing the output:

```
$ curl http://console-openshift-console.apps.<cluster_name>.<base_domain> -I -L --insecure
```

If the configuration is correct, the output from the command shows the following response:

```
HTTP/1.1 302 Found
content-length: 0
location: https://console-openshift-console.apps.<cluster-name>.<base domain>/
cache-control: no-cacheHTTP/1.1 200 OK
referrer-policy: strict-origin-when-cross-origin
set-cookie: csrf-
token=39HoZgztDnzjJkq/JuLJMeoKNXIfiVv2YgZc09c3TBOBU4NI6kDXaJH1LdicNhN1UsQ
Wzon4Dor9GWGfopaTEQ==; Path=/; Secure
x-content-type-options: nosniff
x-dns-prefetch-control: off
x-frame-options: DENY
x-xss-protection: 1; mode=block
date: Tue, 17 Nov 2020 08:42:10 GMT
content-type: text/html; charset=utf-8
set-cookie:
1e2670d92730b515ce3a1bb65da45062=9b714eb87e93cf34853e87a92d6894be; path=/;
HttpOnly; Secure; SameSite=None
cache-control: private
```

d. Verify that you can access each cluster application on port 443, by running the following command and observing the output:

```
$ curl https://console-openshift-console.apps.<cluster_name>.<base_domain> -I -L --insecure
```

If the configuration is correct, the output from the command shows the following response:

```
HTTP/1.1 200 OK
referrer-policy: strict-origin-when-cross-origin
set-cookie: csrf-
token=UIYWOyQ62LWjw2h003xtYSKlh1a0Py2hhctw0WmV2YEdhJjFyQwWcGBsja261dG
LgaYO0nxzVErhiXt6QepA7g==; Path=/; Secure; SameSite=Lax
x-content-type-options: nosniff
x-dns-prefetch-control: off
x-frame-options: DENY
x-xss-protection: 1; mode=block
date: Wed, 04 Oct 2023 16:29:38 GMT
content-type: text/html; charset=utf-8
set-cookie:
1e2670d92730b515ce3a1bb65da45062=1bf5e9573c9a2760c964ed1659cc1673; path=/
HttpOnly; Secure; SameSite=None
cache-control: private
```

2.5.10. Deploying 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

- You have the OpenShift Container Platform installation program and the pull secret for your cluster.
- You have verified that the cloud provider account on your host has the correct permissions to deploy the cluster. An account with incorrect permissions causes the installation process to fail with an error message that displays the missing permissions.
- Optional: Before you create the cluster, you configured an external load balancer in place of the default load balancer.



IMPORTANT

You do not need to specify API and Ingress static addresses for your installation program. If you choose this configuration, you must take additional actions to define network targets that accept an IP address from each referenced vSphere subnet. See the section "Configuring a user-managed load balancer".

Procedure

- In the directory that contains the installation program, initialize the cluster deployment by running the following command:

```
$ ./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**.

Verification

When the cluster deployment completes successfully:

- The terminal displays directions for accessing your cluster, including a link to the web console and credentials for the **kubeadmin** user.
- Credential information also outputs to **<installation_directory>/openshift_install.log**.



IMPORTANT

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

Example output

```
...
INFO Install complete!
INFO To access the cluster as the system:admin user when using 'oc', run 'export
KUBECONFIG=/home/myuser/install_dir/auth/kubeconfig'
INFO Access the OpenShift web-console here: https://console-openshift-
console.apps.mycluster.example.com
INFO Login to the console with user: "kubeadmin", and password: "password"
INFO Time elapsed: 36m22s
```



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.
- It is recommended that you use 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.

2.5.11. Logging in to the cluster by using the CLI

To log in to your cluster as the default system user, export the **kubeconfig** file. This configuration enables the CLI to authenticate and connect to the specific API server created during OpenShift Container Platform installation.

The **kubeconfig** file is specific to a cluster and is created during OpenShift Container Platform installation.

Prerequisites

- You deployed an OpenShift Container Platform cluster.
- You installed the OpenShift CLI (**oc**).

Procedure

1. Export the **kubeadmin** credentials by running the following command:

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

where:

<installation_directory>

Specifies the path to the directory that stores the installation files.

2. Verify you can run **oc** commands successfully using the exported configuration by running the following command:

```
$ oc whoami
```

Example output

```
system:admin
```

2.5.12. Creating registry storage

After you install the cluster, you must create storage for the registry Operator.

2.5.12.1. Image registry removed during installation

On platforms that do not provide shareable object storage, the OpenShift Image Registry Operator bootstraps itself as **Removed**. This allows **openshift-installer** to complete installations on these platform types.

After installation, you must edit the Image Registry Operator configuration to switch the **managementState** from **Removed** to **Managed**. When this has completed, you must configure storage.

2.5.12.2. Image registry storage configuration

The Image Registry Operator is not initially available for platforms that do not provide default storage. After installation, you must configure your registry to use storage so that the Registry Operator is made available.

Configure a persistent volume, which is required for production clusters. Where applicable, you can configure an empty directory as the storage location for non-production clusters.

You can also allow the image registry to use block storage types by using the **Recreate** rollout strategy during upgrades.

2.5.12.2.1. Configuring registry storage for VMware vSphere

As a cluster administrator, following installation you must configure your registry to use storage.

Prerequisites

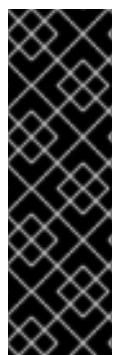
- Cluster administrator permissions.
- A cluster on VMware vSphere.
- Persistent storage provisioned for your cluster, such as Red Hat OpenShift Data Foundation.



IMPORTANT

OpenShift Container Platform supports **ReadWriteOnce** access for image registry storage when you have only one replica. **ReadWriteOnce** access also requires that the registry uses the **Recreate** rollout strategy. To deploy an image registry that supports high availability with two or more replicas, **ReadWriteMany** access is required.

- Must have "100Gi" capacity.



IMPORTANT

Testing shows issues with using the NFS server on RHEL as storage backend for core services. This includes the OpenShift Container Registry and Quay, Prometheus for monitoring storage, and Elasticsearch for logging storage. Therefore, using RHEL NFS to back PVs used by core services is not recommended.

Other NFS implementations on the marketplace might not have these issues. Contact the individual NFS implementation vendor for more information on any testing that was possibly completed against these OpenShift Container Platform core components.

Procedure

1. Change the **spec.storage.pvc** field in the **configs.imageregistry/cluster** resource.



NOTE

When you use shared storage, review your security settings to prevent outside access.

2. Verify that you do not have a registry pod by running the following command:

```
$ oc get pod -n openshift-image-registry -l docker-registry=default
```

Example output

No resources found in openshift-image-registry namespace



NOTE

If you do have a registry pod in your output, you do not need to continue with this procedure.

3. Check the registry configuration by running the following command:

```
$ oc edit configs.imageregistry.operator.openshift.io
```

Example output

```
storage:
pvc:
claim:
```

Leave the **claim** field blank to allow the automatic creation of an **image-registry-storage** persistent volume claim (PVC). The PVC is generated based on the default storage class. However, be aware that the default storage class might provide ReadWriteOnce (RWO) volumes, such as a RADOS Block Device (RBD), which can cause issues when you replicate to more than one replica.

4. Check the **clusteroperator** status by running the following command:

```
$ oc get clusteroperator image-registry
```

Example output

NAME	VERSION	AVAILABLE	PROGRESSING	DEGRADED	SINCE
MESSAGE					
image-registry	4.7	True	False	False	6h50m

2.5.12.2.2. Configuring block registry storage for VMware vSphere

To allow the image registry to use block storage types such as vSphere Virtual Machine Disk (VMDK) during upgrades as a cluster administrator, you can use the **Recreate** rollout strategy.



IMPORTANT

Block storage volumes are supported but not recommended for use with image registry on production clusters. An installation where the registry is configured on block storage is not highly available because the registry cannot have more than one replica.

Procedure

1. Enter the following command to set the image registry storage as a block storage type, patch the registry so that it uses the **Recreate** rollout strategy, and runs with only **1** replica:

```
$ oc patch config.imageregistry.operator.openshift.io/cluster --type=merge -p '{"spec": {"rolloutStrategy": "Recreate", "replicas": 1}}'
```

2. Provision the persistent volume (PV) for the block storage device, and create a persistent volume claim (PVC) for that volume. The requested block volume uses the `ReadWriteOnce` (RWO) access mode.
 - a. Create a **pvc.yaml** file with the following contents to define a VMware vSphere **PersistentVolumeClaim** object:

```
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: image-registry-storage
  namespace: openshift-image-registry
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 100Gi
```

where:

name

Specifies a unique name that represents the **PersistentVolumeClaim** object.

namespace

Specifies the **namespace** for the **PersistentVolumeClaim** object, which is **openshift-image-registry**.

accessModes

Specifies the access mode of the persistent volume claim. With **ReadWriteOnce**, the volume can be mounted with read and write permissions by a single node.

storage

The size of the persistent volume claim.

- b. Enter the following command to create the **PersistentVolumeClaim** object from the file:

```
$ oc create -f pvc.yaml -n openshift-image-registry
```

3. Enter the following command to edit the registry configuration so that it references the correct PVC:

```
$ oc edit config.imageregistry.operator.openshift.io -o yaml
```

Example output

```
storage:
pvc:
claim:
```

By creating a custom PVC, you can leave the **claim** field blank for the default automatic creation of an **image-registry-storage** PVC.

For instructions about configuring registry storage so that it references the correct PVC, see [Configuring the registry for vSphere](#).

2.5.13. Telemetry access for OpenShift Container Platform

To provide metrics about cluster health and the success of updates, the Telemetry service requires internet access. When connected, this service runs automatically by default and registers your cluster to [OpenShift Cluster Manager](#).

After you confirm that your [OpenShift Cluster Manager](#) inventory is correct, either maintained automatically by Telemetry or manually by using OpenShift Cluster Manager, use subscription watch to track your OpenShift Container Platform subscriptions at the account or multi-cluster level. For more information about subscription watch, see "Data Gathered and Used by Red Hat's subscription services" in the *Additional resources* section.

Additional resources

- See [About remote health monitoring](#) for more information about the Telemetry service

2.5.14. Configuring network components to run on the control plane

You can configure networking components to run exclusively on the control plane nodes. By default, OpenShift Container Platform allows any node in the machine config pool to host the **ingressVIP** virtual IP address. However, some environments deploy compute nodes in separate subnets from the control plane nodes, which requires configuring the **ingressVIP** virtual IP address to run on the control plane nodes.



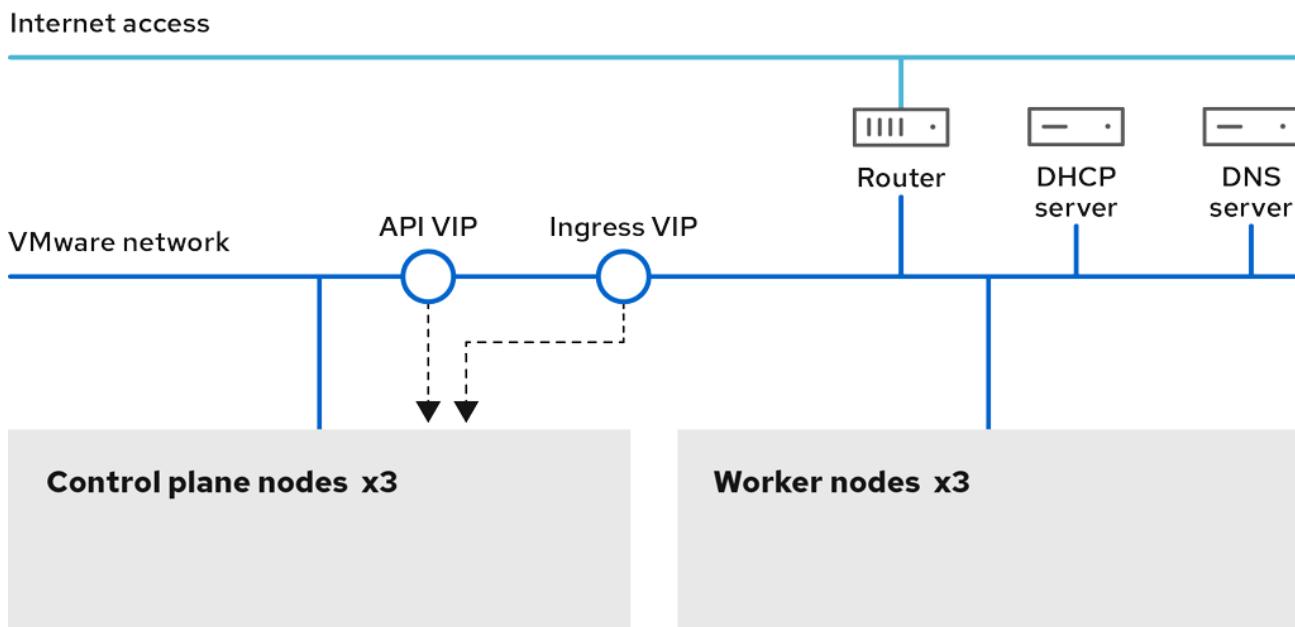
NOTE

You can scale the remote nodes by creating a compute machine set in a separate subnet.



IMPORTANT

When deploying remote nodes in separate subnets, you must place the **ingressVIP** virtual IP address exclusively with the control plane nodes.



Procedure

1. Change to the directory storing the **install-config.yaml** file:

```
$ cd ~/clusterconfigs
```

2. Switch to the **manifests** subdirectory:

```
$ cd manifests
```

3. Create a file named **cluster-network-avoid-workers-99-config.yaml**:

```
$ touch cluster-network-avoid-workers-99-config.yaml
```

4. Open the **cluster-network-avoid-workers-99-config.yaml** file in an editor and enter a custom resource (CR) that describes the Operator configuration:

```
apiVersion: machineconfiguration.openshift.io/v1
kind: MachineConfig
metadata:
  name: 50-worker-fix-ipi-rwn
  labels:
    machineconfiguration.openshift.io/role: worker
spec:
  config:
    ignition:
      version: 3.2.0
    storage:
      files:
        - path: /etc/kubernetes/manifests/keepalived.yaml
          mode: 0644
          contents:
            source: data:,
```

This manifest places the **ingressVIP** virtual IP address on the control plane nodes. Additionally, this manifest deploys the following processes on the control plane nodes only:

- **openshift-ingress-operator**
- **keepalived**

5. Save the **cluster-network-avoid-workers-99-config.yaml** file.
6. Create a **manifests/cluster-ingress-default-ingresscontroller.yaml** file:

```
apiVersion: operator.openshift.io/v1
kind: IngressController
metadata:
  name: default
  namespace: openshift-ingress-operator
spec:
  nodePlacement:
    nodeSelector:
      matchLabels:
        node-role.kubernetes.io/master: ""
```

7. Consider backing up the **manifests** directory. The installer deletes the **manifests/** directory when creating the cluster.
8. Modify the **cluster-scheduler-02-config.yml** manifest to make the control plane nodes schedulable by setting the **mastersSchedulable** field to **true**. Control plane nodes are not schedulable by default. For example:

```
$ sed -i "s;mastersSchedulable: false;mastersSchedulable: true;g"
clusterconfigs/manifests/cluster-scheduler-02-config.yml
```



NOTE

If control plane nodes are not schedulable after completing this procedure, deploying the cluster will fail.

2.5.15. Next steps

- [Customize your cluster.](#)
- [Remote health reporting](#)
- [Set up your registry and configure registry storage](#) .
- Optional: [View the events from the vSphere Problem Detector Operator](#) to determine if the cluster has permission or storage configuration issues.

2.6. INSTALLING A CLUSTER ON VSphere IN A DISCONNECTED ENVIRONMENT

In OpenShift Container Platform 4.19, you can install a cluster on VMware vSphere infrastructure in a restricted network by creating an internal mirror of the installation release content.

2.6.1. Prerequisites

- You have completed the tasks in [Preparing to install a cluster using installer-provisioned infrastructure](#).
- You reviewed your VMware platform licenses. Red Hat does not place any restrictions on your VMware licenses, but some VMware infrastructure components require licensing.
- You reviewed details about the [OpenShift Container Platform installation and update processes](#).
- You read the documentation on [selecting a cluster installation method and preparing it for users](#).
- You [created a registry on your mirror host](#) and obtained the **imageContentSources** data for your version of OpenShift Container Platform.



IMPORTANT

Because the installation media is on the mirror host, you can use that computer to complete all installation steps.

- You provisioned [persistent storage](#) for your cluster. To deploy a private image registry, your storage must provide the ReadWriteMany access mode.
- The OpenShift Container Platform installer requires access to port 443 on the vCenter and ESXi hosts. You verified that port 443 is accessible.
- If you use a firewall, you confirmed with the administrator that port 443 is accessible. Control plane nodes must be able to reach vCenter and ESXi hosts on port 443 for the installation to succeed.
- If you use a firewall and plan to use the Telemetry service, you [configured the firewall to allow the sites](#) that your cluster requires access to.



NOTE

If you are configuring a proxy, be sure to also review this site list.

2.6.2. About installations in restricted networks

In OpenShift Container Platform 4.19, you can perform an installation that does not require an active connection to the internet to obtain software components. Restricted network installations can be completed using installer-provisioned infrastructure or user-provisioned infrastructure, depending on the cloud platform to which you are installing the cluster.

If you choose to perform a restricted network installation on a cloud platform, you still require access to its cloud APIs. Some cloud functions, like Amazon Web Service’s Route 53 DNS and IAM services, require internet access. Depending on your network, you might require less internet access for an installation on bare metal hardware, Nutanix, or on VMware vSphere.

To complete a restricted network installation, you must create a registry that mirrors the contents of the OpenShift image registry and contains the installation media. You can create this registry on a mirror host, which can access both the internet and your closed network, or by using other methods that meet your restrictions.

2.6.2.1. Additional limits

Clusters in restricted networks have the following additional limitations and restrictions:

- The **ClusterVersion** status includes an **Unable to retrieve available updates** error.
- By default, you cannot use the contents of the Developer Catalog because you cannot access the required image stream tags.

2.6.3. Internet access for OpenShift Container Platform

In OpenShift Container Platform 4.19, you require access to the internet to obtain the images that are necessary to install your cluster.

You must have internet access to perform the following actions:

- Access [OpenShift Cluster Manager](#) to download the installation program and perform subscription management. If the cluster has internet access and you do not disable Telemetry, that service automatically entitles your cluster.
- Access [Quay.io](#) to obtain the packages that are required to install your cluster.
- Obtain the packages that are required to perform cluster updates.

2.6.4. Creating the RHCOS image for restricted network installations

Download the Red Hat Enterprise Linux CoreOS (RHCOS) image to install OpenShift Container Platform on a restricted network VMware vSphere environment.

Prerequisites

- Obtain the OpenShift Container Platform installation program. For a restricted network installation, the program is on your mirror registry host.

Procedure

1. Log in to the Red Hat Customer Portal's [Product Downloads page](#).
2. Under **Version**, select the most recent release of OpenShift Container Platform 4.19 for RHEL 8.



IMPORTANT

The RHCOS images might not change with every release of OpenShift Container Platform. You must download images with the highest version that is less than or equal to the OpenShift Container Platform version that you install. Use the image versions that match your OpenShift Container Platform version if they are available.

3. Download the [Red Hat Enterprise Linux CoreOS \(RHCOS\) - vSphere image](#).
4. Upload the image you downloaded to a location that is accessible from the bastion server.

The image is now available for a restricted installation. Note the image name or location for use in OpenShift Container Platform deployment.

2.6.5. VMware vSphere region and zone enablement

You can deploy an OpenShift Container Platform cluster to multiple vSphere data centers. Each data center can run multiple clusters. This configuration reduces the risk of a hardware failure or network outage that can cause your cluster to fail. To enable regions and zones, you must define multiple failure domains for your OpenShift Container Platform cluster.



IMPORTANT

The VMware vSphere region and zone enablement feature requires the vSphere Container Storage Interface (CSI) driver as the default storage driver in the cluster. As a result, the feature is only available on a newly installed cluster.

For a cluster that was upgraded from a previous release, you must enable CSI automatic migration for the cluster. You can then configure multiple regions and zones for the upgraded cluster.

The default installation configuration deploys a cluster to a single vSphere data center. If you want to deploy a cluster to multiple vSphere data centers, you must create an installation configuration file that enables the region and zone feature.

The default **install-config.yaml** file includes **vcenters** and **failureDomains** fields, where you can specify multiple vSphere data centers and clusters for your OpenShift Container Platform cluster. You can leave these fields blank if you want to install an OpenShift Container Platform cluster in a vSphere environment that consists of single data center.

The following list describes terms associated with defining zones and regions for your cluster:

- Failure domain: Establishes the relationships between a region and zone. You define a failure domain by using vCenter objects, such as a **datastore** object. A failure domain defines the vCenter location for OpenShift Container Platform cluster nodes.
- Region: Specifies a vCenter data center. You define a region by using a tag from the **openshift-region** tag category.
- Zone: Specifies a vCenter cluster. You define a zone by using a tag from the **openshift-zone** tag category.



NOTE

If you plan on specifying more than one failure domain in your **install-config.yaml** file, you must create tag categories, zone tags, and region tags in advance of creating the configuration file.

You must create a vCenter tag for each vCenter data center, which represents a region. Additionally, you must create a vCenter tag for each cluster than runs in a data center, which represents a zone. After you create the tags, you must attach each tag to their respective data centers and clusters.

The following table outlines an example of the relationship among regions, zones, and tags for a configuration with multiple vSphere data centers running in a single VMware vCenter.

Data center (region)	Cluster (zone)	Tags
us-east	us-east-1	us-east-1a
		us-east-1b
	us-east-2	us-east-2a
		us-east-2b
us-west	us-west-1	us-west-1a
		us-west-1b
	us-west-2	us-west-2a
		us-west-2b

2.6.6. VMware vSphere host group enablement



IMPORTANT

OpenShift zones support for vSphere host groups is a Technology Preview feature only. Technology Preview features are not supported with Red Hat production service level agreements (SLAs) and might not be functionally complete. Red Hat does not recommend using them in production. These features provide early access to upcoming product features, enabling customers to test functionality and provide feedback during the development process.

For more information about the support scope of Red Hat Technology Preview features, see [Technology Preview Features Support Scope](#).

When deploying an OpenShift Container Platform cluster to VMware vSphere, you can map your vSphere host groups onto OpenShift Container Platform failure domains. This is useful if you are using a stretched cluster configuration, where ESXi hosts are grouped into host groups by physical location.

To enable this feature, you must meet the following requirements:

- You must arrange your ESXi hosts into host groups.
- You must create a vCenter tag in the **openshift-region** tag category for your cluster. After you create the tag, you must attach the tag to the cluster.
- You must create a vCenter tag in the **openshift-zone** tag category for each host group and then attach the correct tag to each ESXi host.
- You must define multiple failure domains for your OpenShift Container Platform cluster in the **install-config.yaml** file.
- You must grant the **Host.Inventory.EditCluster** privilege on the vSphere vCenter cluster object.

- You must include the following parameters in your **install-config.yaml** file to enable this Technology Preview feature:

```
featureSet: TechPreviewNoUpgrade
featureGate:
  - "VSphereHostVMGroupZonal=true"
```



NOTE

For further information on feature gates, see "Enabling features using feature gates".

Review the following key terms, which correspond to parameters in your **install-config.yaml** file that you must configure to enable this feature:

- Failure domain: Specifies the relationships between regions and zones in OpenShift Container Platform, and clusters and host groups in vSphere. You define a failure domain by using vCenter objects, such as a **datastore** object. A failure domain defines the vCenter location for OpenShift Container Platform cluster nodes.
- Region: Specifies a vCenter cluster. You define a region by using a tag from the **openshift-region** tag category.
- Zone: Specifies a vCenter host group. You define a zone by using a tag from the **openshift-zone** tag category.
- Region type: Specifies the **ComputeCluster** region type to enable this feature.
- Zone type: Specifies the **HostGroup** zone type to enable this feature.

Additional resources

- [Additional VMware vSphere configuration parameters](#)
- [Deprecated VMware vSphere configuration parameters](#)
- [vSphere automatic migration](#)
- [VMware vSphere CSI Driver Operator](#)
- [Enabling features using feature gates](#)

2.6.7. Creating the installation configuration file

You can customize the OpenShift Container Platform cluster you install on VMware vSphere.

Prerequisites

- You have the OpenShift Container Platform installation program and the pull secret for your cluster. For a restricted network installation, these files are on your mirror host.

- You have the **imageContentSources** values that were generated during mirror registry creation.
- You have obtained the contents of the certificate for your mirror registry.
- You have retrieved a Red Hat Enterprise Linux CoreOS (RHCOS) image and uploaded it to an accessible location.

Procedure

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

- a. Change to the directory that contains the installation program and run the following command:

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

- **<installation_directory>**: For **<installation_directory>**, specify the directory name to store the files that the installation program creates.
When specifying the directory:
 - Verify that the directory has the **execute** permission. This permission is required to run Terraform binaries under the installation directory.
 - Use an empty directory. Some installation assets, such as bootstrap X.509 certificates, have short expiration intervals, therefore 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, specify an SSH key that your **ssh-agent** process uses.

- ii. Select **vsphere** as the platform to target.
 - iii. Specify the name of your vCenter instance.
 - iv. Specify the user name and password for the vCenter account that has the required permissions to create the cluster.
The installation program connects to your vCenter instance.
 - v. Select the data center in your vCenter instance to connect to.



NOTE

After you create the installation configuration file, you can modify the file to create a multiple vSphere data center environment. This means that you can deploy an OpenShift Container Platform cluster to multiple vSphere data centers. For more information about creating this environment, see the section named *VMware vSphere region and zone enablement*.

- vi. Select the default vCenter datastore to use.



WARNING

You can specify the path of any datastore that exists in a datastore cluster. By default, Storage Distributed Resource Scheduler (SDRS), which uses Storage vMotion, is automatically enabled for a datastore cluster. Red Hat does not support Storage vMotion, so you must disable Storage DRS to avoid data loss issues for your OpenShift Container Platform cluster.

You cannot specify more than one datastore path. If you must specify VMs across multiple datastores, use a **datastore** object to specify a failure domain in your cluster's **install-config.yaml** configuration file. For more information, see "VMware vSphere region and zone enablement".

- vii. Select the vCenter cluster to install the OpenShift Container Platform cluster in. The installation program uses the root resource pool of the vSphere cluster as the default resource pool.
- viii. Select the network in the vCenter instance that contains the virtual IP addresses and DNS records that you configured.
- ix. Enter the virtual IP address that you configured for control plane API access.
- x. Enter the virtual IP address that you configured for cluster ingress.
- xi. Enter the base domain. This base domain must be the same one that you used in the DNS records that you configured.
- xii. Enter a descriptive name for your cluster.
The cluster name you enter must match the cluster name you specified when configuring the DNS records.

2. Choose one of the following methods to specify an RHCOS image for your cluster than runs in a VMware vSphere vCenter environment.

- a. The **clusterOSImage** parameter method: In the **install-config.yaml** file, set the value of **platform.vsphere.clusterOSImage** to the image location or name. For example:

```
platform:
```

vsphere:

```
clusterOSImage: http://mirror.example.com/images/rhcos-43.81.201912131630.0-vmware.x86_64.ova?
sha256=ffebbd68e8a1f2a245ca19522c16c86f67f9ac8e4e0c1f0a812b068b16f7265d
```

b. The **topology.template** parameter method:

- i. Download the **Red Hat Enterprise Linux CoreOS (RHCOS) - vSphere** image in Open Virtual Appliance (OVA) format to your local system. For more information, see "Creating the RHCOS image for restricted network installations".
- ii. From the **Hosts and Clusters** tab on the vSphere Client, right-click your cluster name and select **Deploy OVF Template**.
- iii. On the **Select an OVF** tab, specify the name of the RHCOS OVA file that you downloaded.
- iv. On the **Select a name and folder** tab, set a **Virtual machine name** for your template, such as **Template-RHCOS**.
- v. Click the name of your vSphere cluster and select the folder you created in the previous step.
- vi. On the **Select a compute resource** tab, click the name of your vSphere cluster.
- vii. On the **Select storage** tab, configure the storage options for your VM.
- viii. When creating the OVF template, do not specify values on the **Customize template** tab or configure the template any further.
- ix. In the **install-config.yaml** file, set the value of **topology.template** to the path where you imported the image to your vSphere vCenter instance.

3. Edit the **install-config.yaml** file to give the additional information that is required for an installation in a restricted network.

a. Update the **pullSecret** value to contain the authentication information for your registry:

```
pullSecret: '{"auths": {"<mirror_host_name>:5000": {"auth": "<credentials>","email": "you@example.com"}}}
```

For **<mirror_host_name>**, specify the registry domain name that you specified in the certificate for your mirror registry, and for **<credentials>**, specify the base64-encoded user name and password for your mirror registry.

b. Add the **additionalTrustBundle** parameter and value.

```
additionalTrustBundle: |
-----BEGIN CERTIFICATE-----
ZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZ
-----END CERTIFICATE-----
```

The value must be the contents of the certificate file that you used for your mirror registry. The certificate file can be an existing, trusted certificate authority, or the self-signed certificate that you generated for the mirror registry.

c. Add the image content resources, which resemble the following YAML excerpt:

```
imageContentSources:
- mirrors:
  - <mirror_host_name>:5000/<repo_name>/release
    source: quay.io/openshift-release-dev/ocp-release
- mirrors:
  - <mirror_host_name>:5000/<repo_name>/release
    source: registry.redhat.io/ocp/release
```

For these values, use the **imageContentSources** that you recorded during mirror registry creation.

d. Set the publishing strategy to **Internal**:

```
publish: Internal
```

By setting this option, you create an internal Ingress Controller and a private load balancer.

4. Make any other modifications to the **install-config.yaml** file that you require.

For more information about the parameters, see "Installation configuration parameters".

5. 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.

Additional resources

- [Installation configuration parameters](#)

2.6.7.1. Sample `install-config.yaml` file for an installer-provisioned VMware vSphere cluster

You can customize the **install-config.yaml** file to specify more details about your OpenShift Container Platform cluster's platform or change the values of the required parameters.



NOTE

The sample **install-config.yaml** file shows the **clusterOSImage** parameter to specify the URL for the Red Hat Enterprise Linux CoreOS (RHCOS) image. As an alternative to this configuration, you can use the **topology.template** parameter to point to the path in your vCenter environment that includes an RHCOS image in Open Virtual Appliance (OVA) format.

```
apiVersion: v1
baseDomain: example.com 1
compute: 2
- architecture: amd64
  name: <worker_node>
  platform: {}
replicas: 3
```


- 1 The base domain of the cluster. All DNS records must be sub-domains of this base and include the cluster name.
- 2 3 The **controlPlane** section is a single mapping, but the **compute** section is a sequence of mappings. To meet the requirements of the different data structures, the first line of the **compute** section must begin with a hyphen, -, and the first line of the **controlPlane** section must not. Only one control plane pool is used.
- 4 The cluster name that you specified in your DNS records.
- 5 Optional: Provides additional configuration for the machine pool parameters for the compute and control plane machines.



IMPORTANT

The VIPs, **apiVIP** and **ingressVIP**, must come from the same **networking.machineNetwork** segment. For **apiVIP** and for **ingressVIP**, if the **networking.machineNetwork** is **10.0.0.0/16** then API VIPs and Ingress VIPs must be in one of the **10.0.0.0/16** machine networks.

- 6 Establishes the relationships between a region and zone. You define a failure domain by using vCenter objects, such as a **datastore** object. A failure domain defines the vCenter location for OpenShift Container Platform cluster nodes.
- 7 The path to the vSphere datastore that holds virtual machine files, templates, and ISO images.



IMPORTANT

You can specify the path of any datastore that exists in a datastore cluster. By default, Storage vMotion is automatically enabled for a datastore cluster. Red Hat does not support Storage vMotion, so you must disable Storage vMotion to avoid data loss issues for your OpenShift Container Platform cluster.

If you must specify VMs across multiple datastores, use a **datastore** object to specify a failure domain in your cluster's **install-config.yaml** configuration file. For more information, see "VMware vSphere region and zone enablement".

- 8 Optional: Provides an existing resource pool for machine creation. If you do not specify a value, the installation program uses the root resource pool of the vSphere cluster.
- 9 Optional: Each VM created by OpenShift Container Platform is assigned a unique tag that is specific to the cluster. The assigned tag enables the installation program to identify and remove the associated VMs when a cluster is decommissioned. You can list up to ten additional tag IDs to be attached to the VMs provisioned by the installation program.
- 10 The ID of the tag to be associated by the installation program. For example, **urn:vmomi:InventoryServiceTag:208e713c-cae3-4b7f-918e-4051ca7d1f97:GLOBAL**. For more information about determining the tag ID, see the [vSphere Tags and Attributes documentation](#).
- 11 The vSphere disk provisioning method.
- 12 The location of the Red Hat Enterprise Linux CoreOS (RHCOS) image that is accessible from the bastion server.

- 13 For **<local_registry>**, specify the registry domain name, and optionally the port, that your mirror registry uses to serve content. For example **registry.example.com** or
- 14 Provide the contents of the certificate file that you used for your mirror registry.
- 15 Provide the **imageContentSources** section from the output of the command to mirror the repository.



NOTE

In OpenShift Container Platform 4.12 and later, the **apiVIP** and **ingressVIP** configuration settings are deprecated. Instead, use a list format to enter values in the **apiVIPs** and **ingressVIPs** configuration settings.

2.6.7.2. Configuring the cluster-wide proxy during installation

To enable internet access in environments that deny direct connections, configure a cluster-wide proxy in the **install-config.yaml** file. This configuration ensures that the new OpenShift Container Platform cluster routes traffic through the specified HTTP or HTTPS proxy.

Prerequisites

- You have an existing **install-config.yaml** file.
- You have reviewed the sites that your cluster requires access to and determined whether any of them need to bypass the proxy. By default, all cluster egress traffic is proxied, including calls to hosting cloud provider APIs. You added sites to the **Proxy** object's **spec.noProxy** field to bypass the proxy if necessary.



NOTE

The **Proxy** object **status.noProxy** field is populated with the values of the **networking.machineNetwork[].cidr**, **networking.clusterNetwork[].cidr**, and **networking.serviceNetwork[]** fields from your installation configuration.

For installations on Amazon Web Services (AWS), Google Cloud, Microsoft Azure, and Red Hat OpenStack Platform (RHOSP), the **Proxy** object **status.noProxy** field is also populated with the instance metadata endpoint (169.254.169.254).

Procedure

- 1 Edit your **install-config.yaml** file and add the proxy settings. For example:

```
apiVersion: v1
baseDomain: my.domain.com
proxy:
  httpProxy: http://<username>:<pswd>@<ip>:<port>
  httpsProxy: https://<username>:<pswd>@<ip>:<port>
  noProxy: example.com
  additionalTrustBundle: |
    -----BEGIN CERTIFICATE-----
    <MY_TRUSTED_CA_CERT>
```

```
-----END CERTIFICATE-----
additionalTrustBundlePolicy: <policy_to_add_additionalTrustBundle>
# ...
```

where:

proxy.httpProxy

Specifies a proxy URL to use for creating HTTP connections outside the cluster. The URL scheme must be **http**.

proxy.httpsProxy

Specifies a proxy URL to use for creating HTTPS connections outside the cluster.

proxy.noProxy

Specifies a comma-separated list of destination domain names, IP addresses, or other network CIDRs to exclude from proxying. Preface a domain with `.` to match subdomains only. For example, `.y.com` matches `x.y.com`, but not `y.com`. Use `*` to bypass the proxy for all destinations. You must include vCenter's IP address and the IP range that you use for its machines.

additionalTrustBundle

If provided, the installation program generates a config map that is named **user-ca-bundle** in the **openshift-config** namespace to hold the additional CA certificates. If you provide **additionalTrustBundle** and at least one proxy setting, the **Proxy** object is configured to reference the **user-ca-bundle** config map in the **trustedCA** field. The Cluster Network Operator then creates a **trusted-ca-bundle** config map that merges the contents specified for the **trustedCA** parameter with the RHCOS trust bundle. The **additionalTrustBundle** field is required unless the proxy's identity certificate is signed by an authority from the RHCOS trust bundle.

additionalTrustBundlePolicy

Specifies the policy that determines the configuration of the **Proxy** object to reference the **user-ca-bundle** config map in the **trustedCA** field. The allowed values are **Proxyonly** and **Always**. Use **Proxyonly** to reference the **user-ca-bundle** config map only when **http/https** proxy is configured. Use **Always** to always reference the **user-ca-bundle** config map. The default value is **Proxyonly**. Optional parameter.



NOTE

The installation program does not support the proxy **readinessEndpoints** field.



NOTE

If the installer times out, restart and then complete the deployment by using the **wait-for** command of the installer. For example:

+

```
$ ./openshift-install wait-for install-complete --log-level debug
```

2. Save the file and reference it when installing OpenShift Container Platform.

The installation program creates a cluster-wide proxy that is named **cluster** that uses the proxy settings in the provided **install-config.yaml** file. If no proxy settings are provided, a **cluster Proxy** object is still created, but it will have a nil **spec**.



NOTE

Only the **Proxy** object named **cluster** is supported, and no additional proxies can be created.

2.6.7.3. Configuring regions and zones for a VMware vCenter

You can modify the default installation configuration file, so that you can deploy an OpenShift Container Platform cluster to multiple vSphere data centers.

The default **install-config.yaml** file configuration from the previous release of OpenShift Container Platform is deprecated. You can continue to use the deprecated default configuration, but the **openshift-installer** will prompt you with a warning message that indicates the use of deprecated fields in the configuration file.

Prerequisites

- You have an existing **install-config.yaml** installation configuration file.



IMPORTANT

You must specify at least one failure domain for your OpenShift Container Platform cluster, so that you can provision data center objects for your VMware vCenter server. Consider specifying multiple failure domains if you need to provision virtual machine nodes in different data centers, clusters, datastores, and other components. To enable regions and zones, you must define multiple failure domains for your OpenShift Container Platform cluster.

- You have installed the **govc** command line tool.



IMPORTANT

The example uses the **govc** command. The **govc** command is an open source command available from VMware; it is not available from Red Hat. The Red Hat support team does not maintain the **govc** command. Instructions for downloading and installing **govc** are found on the VMware documentation website.

Procedure

1. Create the **openshift-region** and **openshift-zone** vCenter tag categories by running the following commands:



IMPORTANT

If you specify different names for the **openshift-region** and **openshift-zone** vCenter tag categories, the installation of the OpenShift Container Platform cluster fails.

- \$ govc tags.category.create -d "OpenShift region" openshift-region
- \$ govc tags.category.create -d "OpenShift zone" openshift-zone
- 2. For each region where you want to deploy your cluster, create a region tag by running the following command:

```
  $ govc tags.create -c <region_tag_category> <region_tag>
```

- 3. For each zone where you want to deploy your cluster, create a zone tag by running the following command:
- \$ govc tags.create -c <zone_tag_category> <zone_tag>
- 4. Attach region tags to each vCenter data center object by running the following command:

```
  $ govc tags.attach -c <region_tag_category> <region_tag_1> /<data_center_1>
```

- 5. Attach the zone tags to each vCenter cluster object by running the following command:
- \$ govc tags.attach -c <zone_tag_category> <zone_tag_1> /<data_center_1>/host/<cluster1>
- 6. Change to the directory that contains the installation program and initialize the cluster deployment according to your chosen installation requirements.

Sample `install-config.yaml` file with multiple data centers defined in a vSphere center

```
# ...
compute:
---
vsphere:
  zones:
    - "<machine_pool_zone_1>"
    - "<machine_pool_zone_2>"
# ...
controlPlane:
# ...
vsphere:
  zones:
    - "<machine_pool_zone_1>"
    - "<machine_pool_zone_2>"
# ...
platform:
  vsphere:
    vcenters:
# ...
  datacenters:
    - <data_center_1_name>
    - <data_center_2_name>
failureDomains:
  - name: <machine_pool_zone_1>
    region: <region_tag_1>
    zone: <zone_tag_1>
```

```

server: <fully_qualified_domain_name>
topology:
  datacenter: <data_center_1>
  computeCluster: "/<data_center_1>/host/<cluster1>"
  networks:
    - <VM_Network1_name>
  datastore: "/<data_center_1>/datastore/<datastore1>"
  resourcePool: "/<data_center_1>/host/<cluster1>/Resources/<resourcePool1>"
  folder: "/<data_center_1>/vm/<folder1>"
- name: <machine_pool_zone_2>
  region: <region_tag_2>
  zone: <zone_tag_2>
  server: <fully_qualified_domain_name>
  topology:
    datacenter: <data_center_2>
    computeCluster: "/<data_center_2>/host/<cluster2>"
    networks:
      - <VM_Network2_name>
    datastore: "/<data_center_2>/datastore/<datastore2>"
    resourcePool: "/<data_center_2>/host/<cluster2>/Resources/<resourcePool2>"
    folder: "/<data_center_2>/vm/<folder2>"
# ...

```

2.6.7.4. Configuring host groups for a VMware vCenter



IMPORTANT

OpenShift zones support for vSphere host groups is a Technology Preview feature only. Technology Preview features are not supported with Red Hat production service level agreements (SLAs) and might not be functionally complete. Red Hat does not recommend using them in production. These features provide early access to upcoming product features, enabling customers to test functionality and provide feedback during the development process.

For more information about the support scope of Red Hat Technology Preview features, see [Technology Preview Features Support Scope](#).

You can modify the default installation configuration file to deploy an OpenShift Container Platform cluster on a VMware vSphere stretched cluster, where ESXi hosts are grouped into host groups by physical location.

The default **install-config.yaml** file configuration from previous releases of OpenShift Container Platform is deprecated. Though you can still use it, the OpenShift Container Platform installer will display a warning message that indicates the use of deprecated fields in the configuration file.

Prerequisites

- You have an existing **install-config.yaml** installation configuration file.
- You have arranged your ESXi hosts into host groups.
- You have granted the **Host.Inventory.EditCluster** privilege on the vSphere vCenter cluster object.

- You have downloaded and installed the **govc** command line tool. Instructions can be found on the VMware documentation website. Note that **govc** is an open-source tool that is not maintained by the Red Hat support team.
- You have enabled the **TechPreviewNoUpgrade** feature set. For more information, see "Enabling features using feature gates".



IMPORTANT

To enable host group support, you must define multiple failure domains for your OpenShift Container Platform cluster.

Procedure

1. Create the **openshift-region** and **openshift-zone** vCenter tag categories by running the following commands:



IMPORTANT

If you specify different names for the **openshift-region** and **openshift-zone** vCenter tag categories, the installation of the OpenShift Container Platform cluster fails.

```
$ govc tags.category.create -d "OpenShift region" openshift-region
```

```
$ govc tags.category.create -d "OpenShift zone" openshift-zone
```

2. Create a region tag for the vSphere cluster where you want to deploy your OpenShift Container Platform cluster by entering the following command:

```
$ govc tags.create -c <region_tag_category> <region_tag>
```

3. Create a zone tag for each host group by entering the following command as needed:

```
$ govc tags.create -c <zone_tag_category> <zone_tag>
```

4. Attach the region tag to the vCenter cluster object by entering the following command:

```
$ govc tags.attach -c <region_tag_category> <region_tag_1>
/<datacenter_1>/host/<cluster_1>
```

5. Use zone tags to associate each ESXi host with its host group, by entering the following command for each ESXi host:

```
$ govc tags.attach -c <zone_tag_category> <zone_tag_for_host_group_1>
/<datacenter_1>/host/<cluster_1>/<esxi_host_in_host_group_1>
```

6. Change to the directory that contains the installation program and initialize the cluster deployment according to your chosen installation requirements.

Sample `install-config.yaml` file with multiple host groups



```

featureSet: TechPreviewNoUpgrade
featureGate:
  - "VSphereHostVMGroupZonal=true"
# ...
platform:
  vsphere:
    vcenters:
# ...
    datacenters:
      - <data_center_1_name>
failureDomains:
  - name: <host_group_1>
    region: <cluster_1_region_tag>
    zone: <host_group_1_zone_tag>
    regionType: "ComputeCluster"
    zoneType: "HostGroup"
    server: <fully_qualified_domain_name>
    topology:
      datacenter: <data_center_1>
      computeCluster: "/<data_center_1>/host/<cluster_1>"
      networks:
        - <VM_Network1_name>
        hostGroup: <host_group_1_name>
        datastore: "/<data_center_1>/datastore/<datastore_1>"
        resourcePool: "/<data_center_1>/host/<cluster_1>/Resources/<resourcePool_1>"
        folder: "/<data_center_1>/vm/<folder_1>"
  - name: <host_group_2>
    region: <cluster_1_region_tag>
    zone: <host_group_2_zone_tag>
    regionType: "ComputeCluster"
    zoneType: "HostGroup"
    server: <fully_qualified_domain_name>
    topology:
      datacenter: <data_center_1>
      computeCluster: "/<data_center_1>/host/<cluster_1>"
      networks:
        - <VM_Network1_name>
        hostGroup: <host_group_2_name>
        datastore: "/<data_center_1>/datastore/<datastore_1>"
        resourcePool: "/<data_center_1>/host/<cluster_1>/Resources/<resourcePool_1>"
        folder: "/<data_center_1>/vm/<folder_1>"
```

Additional resources

- [Enabling features using feature gates](#)

2.6.8. Services for a user-managed load balancer

To integrate your infrastructure with existing network standards or gain more control over traffic management in OpenShift Container Platform, configure services for a user-managed load balancer.



IMPORTANT

Configuring a user-managed load balancer depends on your vendor's load balancer.

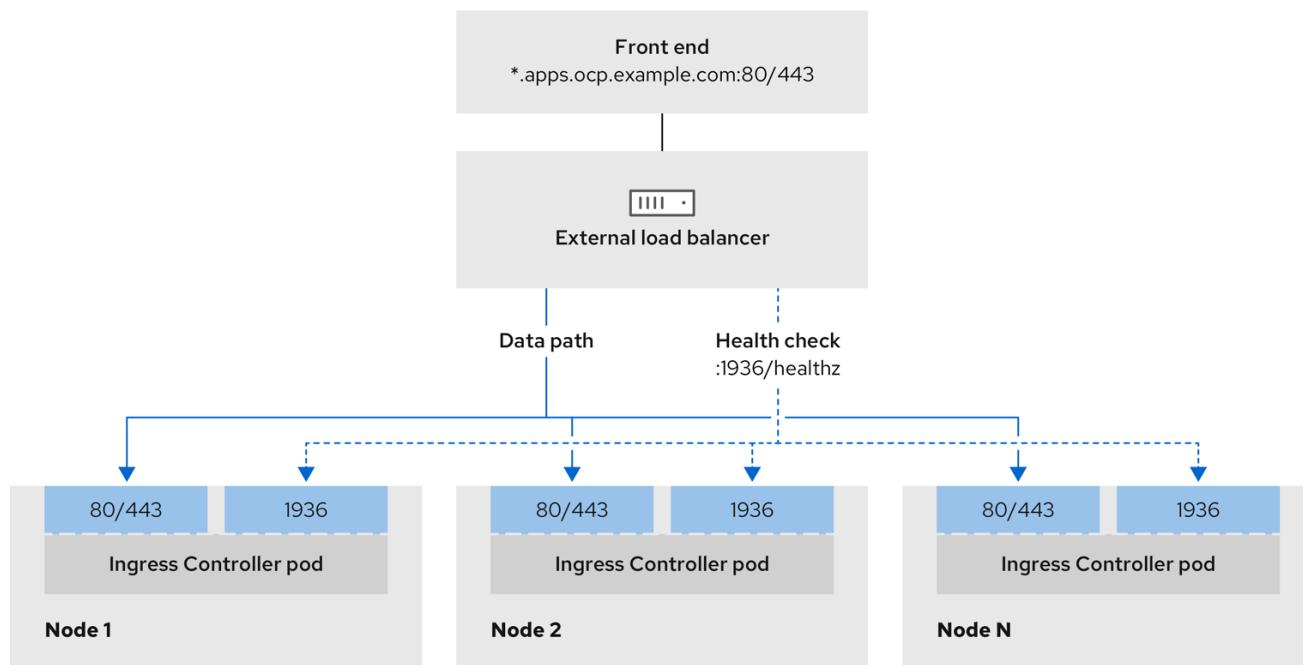
The information and examples in this section are for guideline purposes only. Consult the vendor documentation for more specific information about the vendor's load balancer.

Red Hat supports the following services for a user-managed load balancer:

- Ingress Controller
- OpenShift API
- OpenShift MachineConfig API

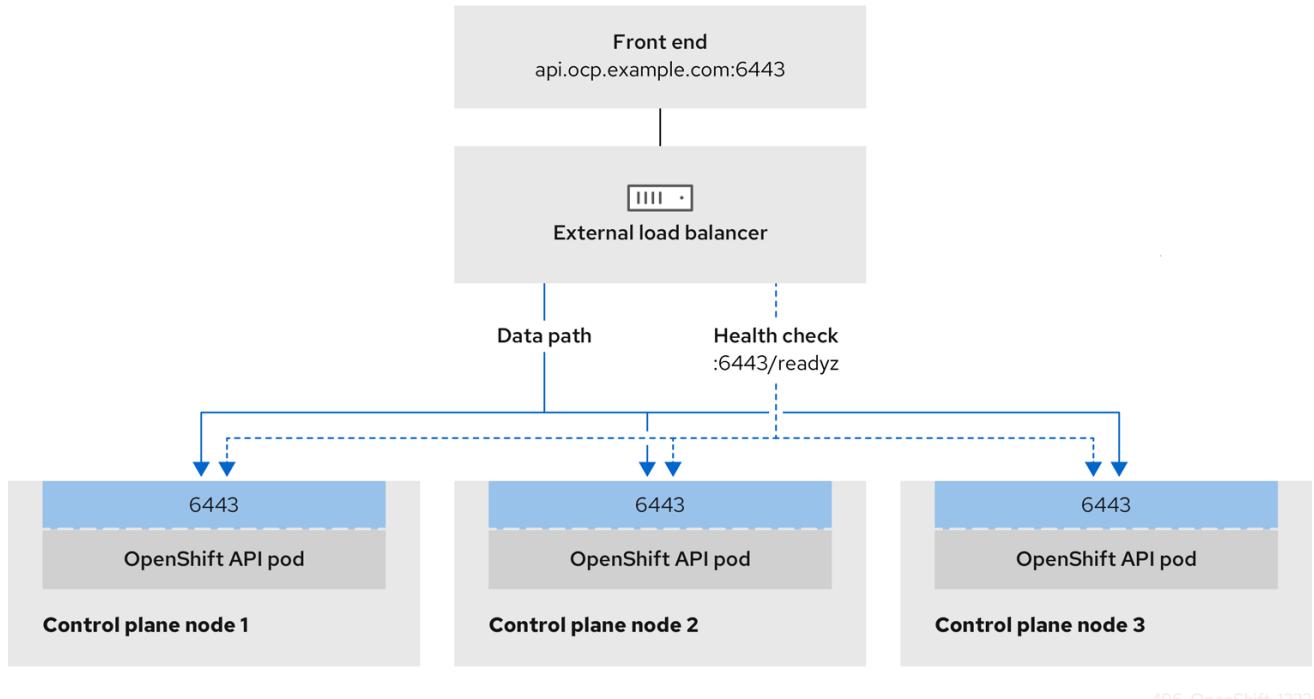
You can choose whether you want to configure one or all of these services for a user-managed load balancer. Configuring only the Ingress Controller service is a common configuration option. To better understand each service, view the following diagrams:

Figure 2.7. Example network workflow that shows an Ingress Controller operating in an OpenShift Container Platform environment



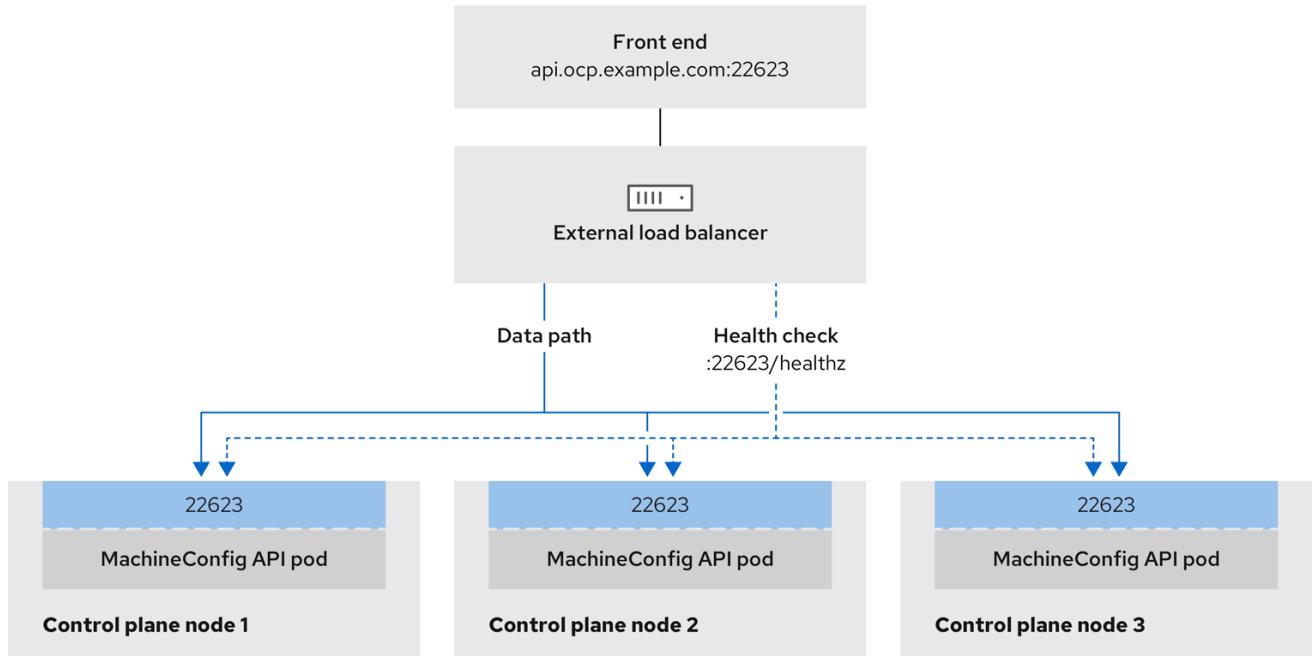
496_OpenShift_1223

Figure 2.8. Example network workflow that shows an OpenShift API operating in an OpenShift Container Platform environment



496_OpenShift_I223

Figure 2.9. Example network workflow that shows an OpenShift MachineConfig API operating in an OpenShift Container Platform environment



496_OpenShift_I223

The following configuration options are supported for user-managed load balancers:

- Use a node selector to map the Ingress Controller to a specific set of nodes. You must assign a static IP address to each node in this set, or configure each node to receive the same IP address from the Dynamic Host Configuration Protocol (DHCP). Infrastructure nodes commonly receive this type of configuration.

- Target all IP addresses on a subnet. This configuration can reduce maintenance overhead, because you can create and destroy nodes within those networks without reconfiguring the load balancer targets. If you deploy your ingress pods by using a machine set on a smaller network, such as a /27 or /28, you can simplify your load balancer targets.

TIP

You can list all IP addresses that exist in a network by checking the machine config pool's resources.

Before you configure a user-managed load balancer for your OpenShift Container Platform cluster, consider the following information:

- For a front-end IP address, you can use the same IP address for the front-end IP address, the Ingress Controller's load balancer, and API load balancer. Check the vendor's documentation for this capability.
- For a back-end IP address, ensure that an IP address for an OpenShift Container Platform control plane node does not change during the lifetime of the user-managed load balancer. You can achieve this by completing one of the following actions:
 - Assign a static IP address to each control plane node.
 - Configure each node to receive the same IP address from the DHCP every time the node requests a DHCP lease. Depending on the vendor, the DHCP lease might be in the form of an IP reservation or a static DHCP assignment.
- Manually define each node that runs the Ingress Controller in the user-managed load balancer for the Ingress Controller back-end service. For example, if the Ingress Controller moves to an undefined node, a connection outage can occur.

2.6.8.1. Configuring a user-managed load balancer

To integrate your infrastructure with existing network standards or gain more control over traffic management in OpenShift Container Platform, use a user-managed load balancer in place of the default load balancer.



IMPORTANT

Before you configure a user-managed load balancer, ensure that you read the "Services for a user-managed load balancer" section.

Read the following prerequisites that apply to the service that you want to configure for your user-managed load balancer.



NOTE

MetalLB, which runs on a cluster, functions as a user-managed load balancer.

Prerequisites

The following list details OpenShift API prerequisites:

- You defined a front-end IP address.

- TCP ports 6443 and 22623 are exposed on the front-end IP address of your load balancer. Check the following items:
 - Port 6443 provides access to the OpenShift API service.
 - Port 22623 can provide ignition startup configurations to nodes.
- The front-end IP address and port 6443 are reachable by all users of your system with a location external to your OpenShift Container Platform cluster.
- The front-end IP address and port 22623 are reachable only by OpenShift Container Platform nodes.
- The load balancer backend can communicate with OpenShift Container Platform control plane nodes on port 6443 and 22623.

The following list details Ingress Controller prerequisites:

- You defined a front-end IP address.
- TCP port 443 and port 80 are exposed on the front-end IP address of your load balancer.
- The front-end IP address, port 80 and port 443 are reachable by all users of your system with a location external to your OpenShift Container Platform cluster.
- The front-end IP address, port 80 and port 443 are reachable by all nodes that operate in your OpenShift Container Platform cluster.
- The load balancer backend can communicate with OpenShift Container Platform nodes that run the Ingress Controller on ports 80, 443, and 1936.

The following list details prerequisites for health check URL specifications:

You can configure most load balancers by setting health check URLs that determine if a service is available or unavailable. OpenShift Container Platform provides these health checks for the OpenShift API, Machine Configuration API, and Ingress Controller backend services.

The following example shows a Kubernetes API health check specification for a backend service:

```
Path: HTTPS:6443/readyz
Healthy threshold: 2
Unhealthy threshold: 2
Timeout: 10
Interval: 10
```

The following example shows a Machine Config API health check specification for a backend service:

```
Path: HTTPS:22623/healthz
Healthy threshold: 2
Unhealthy threshold: 2
Timeout: 10
Interval: 10
```

The following example shows a Ingress Controller health check specification for a backend service:

```
Path: HTTP:1936/healthz/ready
```

```
Healthy threshold: 2
Unhealthy threshold: 2
Timeout: 5
Interval: 10
```

Procedure

1. Configure the HAProxy Ingress Controller, so that you can enable access to the cluster from your load balancer on ports 6443, 22623, 443, and 80. Depending on your needs, you can specify the IP address of a single subnet or IP addresses from multiple subnets in your HAProxy configuration.

Example HAProxy configuration with one listed subnet

```
# ...
listen my-cluster-api-6443
  bind 192.168.1.100:6443
  mode tcp
  balance roundrobin
  option httpchk
  http-check connect
  http-check send meth GET uri /readyz
  http-check expect status 200
  server my-cluster-master-2 192.168.1.101:6443 check inter 10s rise 2 fall 2
  server my-cluster-master-0 192.168.1.102:6443 check inter 10s rise 2 fall 2
  server my-cluster-master-1 192.168.1.103:6443 check inter 10s rise 2 fall 2

listen my-cluster-machine-config-api-22623
  bind 192.168.1.100:22623
  mode tcp
  balance roundrobin
  option httpchk
  http-check connect
  http-check send meth GET uri /healthz
  http-check expect status 200
  server my-cluster-master-2 192.168.1.101:22623 check inter 10s rise 2 fall 2
  server my-cluster-master-0 192.168.1.102:22623 check inter 10s rise 2 fall 2
  server my-cluster-master-1 192.168.1.103:22623 check inter 10s rise 2 fall 2

listen my-cluster-apps-443
  bind 192.168.1.100:443
  mode tcp
  balance roundrobin
  option httpchk
  http-check connect
  http-check send meth GET uri /healthz/ready
  http-check expect status 200
  server my-cluster-worker-0 192.168.1.111:443 check port 1936 inter 10s rise 2 fall 2
  server my-cluster-worker-1 192.168.1.112:443 check port 1936 inter 10s rise 2 fall 2
  server my-cluster-worker-2 192.168.1.113:443 check port 1936 inter 10s rise 2 fall 2

listen my-cluster-apps-80
  bind 192.168.1.100:80
  mode tcp
  balance roundrobin
  option httpchk
```

```

http-check connect
http-check send meth GET uri /healthz/ready
http-check expect status 200
  server my-cluster-worker-0 192.168.1.111:80 check port 1936 inter 10s rise 2 fall 2
  server my-cluster-worker-1 192.168.1.112:80 check port 1936 inter 10s rise 2 fall 2
  server my-cluster-worker-2 192.168.1.113:80 check port 1936 inter 10s rise 2 fall 2
# ...

```

Example HAProxy configuration with multiple listed subnets

```

# ...
listen api-server-6443
  bind *:6443
  mode tcp
    server master-00 192.168.83.89:6443 check inter 1s
    server master-01 192.168.84.90:6443 check inter 1s
    server master-02 192.168.85.99:6443 check inter 1s
    server bootstrap 192.168.80.89:6443 check inter 1s

listen machine-config-server-22623
  bind *:22623
  mode tcp
    server master-00 192.168.83.89:22623 check inter 1s
    server master-01 192.168.84.90:22623 check inter 1s
    server master-02 192.168.85.99:22623 check inter 1s
    server bootstrap 192.168.80.89:22623 check inter 1s

listen ingress-router-80
  bind *:80
  mode tcp
  balance source
    server worker-00 192.168.83.100:80 check inter 1s
    server worker-01 192.168.83.101:80 check inter 1s

listen ingress-router-443
  bind *:443
  mode tcp
  balance source
    server worker-00 192.168.83.100:443 check inter 1s
    server worker-01 192.168.83.101:443 check inter 1s

listen ironic-api-6385
  bind *:6385
  mode tcp
  balance source
    server master-00 192.168.83.89:6385 check inter 1s
    server master-01 192.168.84.90:6385 check inter 1s
    server master-02 192.168.85.99:6385 check inter 1s
    server bootstrap 192.168.80.89:6385 check inter 1s

listen inspector-api-5050
  bind *:5050
  mode tcp
  balance source
    server master-00 192.168.83.89:5050 check inter 1s
    server master-01 192.168.84.90:5050 check inter 1s

```

```
server master-02 192.168.85.99:5050 check inter 1s
server bootstrap 192.168.80.89:5050 check inter 1s
# ...
```

2. Use the **curl** CLI command to verify that the user-managed load balancer and its resources are operational:

- Verify that the cluster machine configuration API is accessible to the Kubernetes API server resource, by running the following command and observing the response:

```
$ curl https://<loadbalancer_ip_address>:6443/version --insecure
```

If the configuration is correct, you receive a JSON object in response:

```
{
  "major": "1",
  "minor": "11+",
  "gitVersion": "v1.11.0+ad103ed",
  "gitCommit": "ad103ed",
  "gitTreeState": "clean",
  "buildDate": "2019-01-09T06:44:10Z",
  "goVersion": "go1.10.3",
  "compiler": "gc",
  "platform": "linux/amd64"
}
```

- Verify that the cluster machine configuration API is accessible to the Machine config server resource, by running the following command and observing the output:

```
$ curl -v https://<loadbalancer_ip_address>:22623/healthz --insecure
```

If the configuration is correct, the output from the command shows the following response:

```
HTTP/1.1 200 OK
Content-Length: 0
```

- Verify that the controller is accessible to the Ingress Controller resource on port 80, by running the following command and observing the output:

```
$ curl -I -L -H "Host: console-openshift-console.apps.<cluster_name>.<base_domain>" 
http://<load_balancer_front_end_IP_address>
```

If the configuration is correct, the output from the command shows the following response:

```
HTTP/1.1 302 Found
content-length: 0
location: https://console-openshift-console.apps.ocp4.private.opequon.net/
cache-control: no-cache
```

- Verify that the controller is accessible to the Ingress Controller resource on port 443, by running the following command and observing the output:

```
$ curl -I -L --insecure --resolve console-openshift-console.apps.<cluster_name>.<base_domain>:443:<Load Balancer Front End IP Address> https://console-openshift-console.apps.<cluster_name>.<base_domain>
```

If the configuration is correct, the output from the command shows the following response:

```
HTTP/1.1 200 OK
referrer-policy: strict-origin-when-cross-origin
set-cookie: csrf-
token=UIYWOyQ62LWjw2h003xtYSKlh1a0Py2hhctw0WmV2YEdhJjFyQwWcGBsja261dG
LgaYO0nxzVERhiXt6QepA7g==; Path=/; Secure; SameSite=Lax
x-content-type-options: nosniff
x-dns-prefetch-control: off
x-frame-options: DENY
x-xss-protection: 1; mode=block
date: Wed, 04 Oct 2023 16:29:38 GMT
content-type: text/html; charset=utf-8
set-cookie:
1e2670d92730b515ce3a1bb65da45062=1bf5e9573c9a2760c964ed1659cc1673; path=/
HttpOnly; Secure; SameSite=None
cache-control: private
```

3. Configure the DNS records for your cluster to target the front-end IP addresses of the user-managed load balancer. You must update records to your DNS server for the cluster API and applications over the load balancer. The following examples shows modified DNS records:

<load_balancer_ip_address> [A](#) api.<cluster_name>.<base_domain>
 A record pointing to Load Balancer Front End

<load_balancer_ip_address> [A](#) apps.<cluster_name>.<base_domain>
 A record pointing to Load Balancer Front End



IMPORTANT

DNS propagation might take some time for each DNS record to become available. Ensure that each DNS record propagates before validating each record.

4. For your OpenShift Container Platform cluster to use the user-managed load balancer, you must specify the following configuration in your cluster's **install-config.yaml** file:

```
# ...
platform:
  vsphere:
    loadBalancer:
      type: UserManaged
    apiVIPs:
      - <api_ip> 1
    ingressVIPs:
      - <ingress_ip> 2
# ...
```

where:

loadBalancer.type

Set **UserManaged** for the **type** parameter to specify a user-managed load balancer for your cluster. The parameter defaults to **OpenShiftManagedDefault**, which denotes the default internal load balancer. For services defined in an **openshift-kni-infra** namespace, a user-managed load balancer can deploy the **coredns** service to pods in your cluster but ignores **keepalived** and **haproxy** services.

loadBalancer.<api_ip>

Specifies a user-managed load balancer. Specify the user-managed load balancer's public IP address, so that the Kubernetes API can communicate with the user-managed load balancer. Mandatory parameter.

loadBalancer.<ingress_ip>

Specifies a user-managed load balancer. Specify the user-managed load balancer's public IP address, so that the user-managed load balancer can manage ingress traffic for your cluster. Mandatory parameter.

Verification

1. Use the **curl** CLI command to verify that the user-managed load balancer and DNS record configuration are operational:
 - a. Verify that you can access the cluster API, by running the following command and observing the output:

```
$ curl https://api.<cluster_name>.<base_domain>:6443/version --insecure
```

If the configuration is correct, you receive a JSON object in response:

```
{
  "major": "1",
  "minor": "11+",
  "gitVersion": "v1.11.0+ad103ed",
  "gitCommit": "ad103ed",
  "gitTreeState": "clean",
  "buildDate": "2019-01-09T06:44:10Z",
  "goVersion": "go1.10.3",
  "compiler": "gc",
  "platform": "linux/amd64"
}
```

- b. Verify that you can access the cluster machine configuration, by running the following command and observing the output:

```
$ curl -v https://api.<cluster_name>.<base_domain>:22623/healthz --insecure
```

If the configuration is correct, the output from the command shows the following response:

```
HTTP/1.1 200 OK
Content-Length: 0
```

- c. Verify that you can access each cluster application on port 80, by running the following command and observing the output:

```
$ curl http://console-openshift-console.apps.<cluster_name>.<base_domain> -I -L --insecure
```

If the configuration is correct, the output from the command shows the following response:

```
HTTP/1.1 302 Found
content-length: 0
location: https://console-openshift-console.apps.<cluster-name>.<base domain>/
cache-control: no-cacheHTTP/1.1 200 OK
referrer-policy: strict-origin-when-cross-origin
set-cookie: csrf-
token=39HoZgztDnzjJkq/JuLJMeoKNXIfiVv2YgZc09c3TBOBU4NI6kDXaJH1LdicNhN1UsQ
Wzon4Dor9GWGfopaTEQ==; Path=/; Secure
x-content-type-options: nosniff
x-dns-prefetch-control: off
x-frame-options: DENY
x-xss-protection: 1; mode=block
date: Tue, 17 Nov 2020 08:42:10 GMT
content-type: text/html; charset=utf-8
set-cookie:
1e2670d92730b515ce3a1bb65da45062=9b714eb87e93cf34853e87a92d6894be; path=/;
HttpOnly; Secure; SameSite=None
cache-control: private
```

d. Verify that you can access each cluster application on port 443, by running the following command and observing the output:

```
$ curl https://console-openshift-console.apps.<cluster_name>.<base_domain> -I -L --insecure
```

If the configuration is correct, the output from the command shows the following response:

```
HTTP/1.1 200 OK
referrer-policy: strict-origin-when-cross-origin
set-cookie: csrf-
token=UIYW0yQ62LWjw2h003xtYSKlh1a0Py2hhctw0WmV2YEdhJjFyQwWcGBsja261dGLgaYO0nxzVERhiXt6QepA7g==; Path=/; Secure; SameSite=Lax
x-content-type-options: nosniff
x-dns-prefetch-control: off
x-frame-options: DENY
x-xss-protection: 1; mode=block
date: Wed, 04 Oct 2023 16:29:38 GMT
content-type: text/html; charset=utf-8
set-cookie:
1e2670d92730b515ce3a1bb65da45062=1bf5e9573c9a2760c964ed1659cc1673; path=/;
HttpOnly; Secure; SameSite=None
cache-control: private
```

2.6.9. Deploying 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

- You have the OpenShift Container Platform installation program and the pull secret for your cluster.
- You have verified that the cloud provider account on your host has the correct permissions to deploy the cluster. An account with incorrect permissions causes the installation process to fail with an error message that displays the missing permissions.
- Optional: Before you create the cluster, you configured an external load balancer in place of the default load balancer.



IMPORTANT

You do not need to specify API and Ingress static addresses for your installation program. If you choose this configuration, you must take additional actions to define network targets that accept an IP address from each referenced vSphere subnet. See the section "Configuring a user-managed load balancer".

Procedure

- In the directory that contains the installation program, initialize the cluster deployment by running the following command:

```
$ ./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**.

Verification

When the cluster deployment completes successfully:

- The terminal displays directions for accessing your cluster, including a link to the web console and credentials for the **kubeadmin** user.
- Credential information also outputs to **<installation_directory>/openshift_install.log**.



IMPORTANT

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

Example output

```
...
INFO Install complete!
INFO To access the cluster as the system:admin user when using 'oc', run 'export
KUBECONFIG=/home/myuser/install_dir/auth/kubeconfig'
INFO Access the OpenShift web-console here: https://console-openshift-
console.apps.mycluster.example.com
INFO Login to the console with user: "kubeadmin", and password: "password"
INFO Time elapsed: 36m22s
```



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.
- It is recommended that you use 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.

2.6.10. Logging in to the cluster by using the CLI

To log in to your cluster as the default system user, export the **kubeconfig** file. This configuration enables the CLI to authenticate and connect to the specific API server created during OpenShift Container Platform installation.

The **kubeconfig** file is specific to a cluster and is created during OpenShift Container Platform installation.

Prerequisites

- You deployed an OpenShift Container Platform cluster.
- You installed the OpenShift CLI (**oc**).

Procedure

1. Export the **kubeadm** credentials by running the following command:

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

where:

<installation_directory>

Specifies the path to the directory that stores the installation files.

2. Verify you can run **oc** commands successfully using the exported configuration by running the following command:



```
$ oc whoami
```

Example output

```
system:admin
```

2.6.11. Disabling the default OperatorHub catalog sources

Operator catalogs that source content provided by Red Hat and community projects are configured for OperatorHub by default during an OpenShift Container Platform installation. In a restricted network environment, you must disable the default catalogs as a cluster administrator.

Procedure

- Disable the sources for the default catalogs by adding **disableAllDefaultSources: true** to the **OperatorHub** object:

```
$ oc patch OperatorHub cluster --type json \
-p '[{"op": "add", "path": "/spec/disableAllDefaultSources", "value": true}]'
```

TIP

Alternatively, you can use the web console to manage catalog sources. From the **Administration** → **Cluster Settings** → **Configuration** → **OperatorHub** page, click the **Sources** tab, where you can create, update, delete, disable, and enable individual sources.

2.6.12. Creating registry storage

After you install the cluster, you must create storage for the Registry Operator.

2.6.12.1. Image registry removed during installation

On platforms that do not provide shareable object storage, the OpenShift Image Registry Operator bootstraps itself as **Removed**. This allows **openshift-installer** to complete installations on these platform types.

After installation, you must edit the Image Registry Operator configuration to switch the **managementState** from **Removed** to **Managed**. When this has completed, you must configure storage.

2.6.12.2. Image registry storage configuration

The Image Registry Operator is not initially available for platforms that do not provide default storage. After installation, you must configure your registry to use storage so that the Registry Operator is made available.

Configure a persistent volume, which is required for production clusters. Where applicable, you can configure an empty directory as the storage location for non-production clusters.

You can also allow the image registry to use block storage types by using the **Recreate** rollout strategy during upgrades.

2.6.12.2.1. Configuring registry storage for VMware vSphere

As a cluster administrator, following installation you must configure your registry to use storage.

Prerequisites

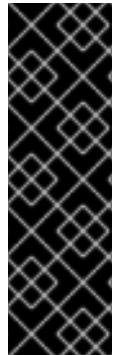
- Cluster administrator permissions.
- A cluster on VMware vSphere.
- Persistent storage provisioned for your cluster, such as Red Hat OpenShift Data Foundation.



IMPORTANT

OpenShift Container Platform supports **ReadWriteOnce** access for image registry storage when you have only one replica. **ReadWriteOnce** access also requires that the registry uses the **Recreate** rollout strategy. To deploy an image registry that supports high availability with two or more replicas, **ReadWriteMany** access is required.

- Must have "100Gi" capacity.



IMPORTANT

Testing shows issues with using the NFS server on RHEL as storage backend for core services. This includes the OpenShift Container Registry and Quay, Prometheus for monitoring storage, and Elasticsearch for logging storage. Therefore, using RHEL NFS to back PVs used by core services is not recommended.

Other NFS implementations on the marketplace might not have these issues. Contact the individual NFS implementation vendor for more information on any testing that was possibly completed against these OpenShift Container Platform core components.

Procedure

1. Change the **spec.storage.pvc** field in the **configs.imageregistry/cluster** resource.



NOTE

When you use shared storage, review your security settings to prevent outside access.

2. Verify that you do not have a registry pod by running the following command:

```
$ oc get pod -n openshift-image-registry -l docker-registry=default
```

Example output

```
No resources found in openshift-image-registry namespace
```



NOTE

If you do have a registry pod in your output, you do not need to continue with this procedure.

3. Check the registry configuration by running the following command:

```
$ oc edit configs.imageregistry.operator.openshift.io
```

Example output

```
storage:
pvc:
claim:
```

Leave the **claim** field blank to allow the automatic creation of an **image-registry-storage** persistent volume claim (PVC). The PVC is generated based on the default storage class. However, be aware that the default storage class might provide ReadWriteOnce (RWO) volumes, such as a RADOS Block Device (RBD), which can cause issues when you replicate to more than one replica.

4. Check the **clusteroperator** status by running the following command:

```
$ oc get clusteroperator image-registry
```

Example output

NAME	VERSION	AVAILABLE	PROGRESSING	DEGRADED	SINCE
MESSAGE					
image-registry	4.7	True	False	False	6h50m

2.6.13. Telemetry access for OpenShift Container Platform

To provide metrics about cluster health and the success of updates, the Telemetry service requires internet access. When connected, this service runs automatically by default and registers your cluster to [OpenShift Cluster Manager](#).

After you confirm that your [OpenShift Cluster Manager](#) inventory is correct, either maintained automatically by Telemetry or manually by using OpenShift Cluster Manager, use subscription watch to track your OpenShift Container Platform subscriptions at the account or multi-cluster level. For more information about subscription watch, see "Data Gathered and Used by Red Hat's subscription services" in the *Additional resources* section.

Additional resources

- See [About remote health monitoring](#) for more information about the Telemetry service

2.6.14. Next steps

- [Customize your cluster](#).
- If necessary, you can [Remote health reporting](#).
- If necessary, see [Registering your disconnected cluster](#).
- [Set up your registry and configure registry storage](#).

CHAPTER 3. USER-PROVISIONED INFRASTRUCTURE

3.1. VSphere Installation Requirements for User-Provisioned Infrastructure

Before you begin an installation on infrastructure that you provision, be sure that your vSphere environment meets the following installation requirements.

3.1.1. VMware vSphere Infrastructure Requirements

You must install an OpenShift Container Platform cluster on one of the following versions of a VMware vSphere instance that meets the requirements for the components that you use:

- Version 8.0 Update 1 or later, or VMware Cloud Foundation 5.0 or later
- VMware vSphere Foundation 9 or later, or VMware Cloud Foundation 9 or later

Both of these releases support Container Storage Interface (CSI) migration, which is enabled by default on OpenShift Container Platform 4.19.

You can host the VMware vSphere infrastructure on-premise or on a [VMware Cloud Verified provider](#) that meets the requirements outlined in the following tables:

Table 3.1. Version requirements for vSphere virtual environments

Virtual environment product	Required version
VMware virtual hardware	15 or later
vSphere ESXi hosts	8.0 Update 1 or later, or VMware vSphere Foundation 9 or later; VMware Cloud Foundation 5.0 or later, or VMware Cloud Foundation 9 or later
vCenter host	8.0 Update 1 or later, or VMware vSphere Foundation 9 or later; VMware Cloud Foundation 5.0 or later, or VMware Cloud Foundation 9 or later



IMPORTANT

You must ensure that the time on your ESXi hosts is synchronized before you install OpenShift Container Platform. See [Edit Time Configuration for a Host](#) in the VMware documentation.

Table 3.2. Minimum supported vSphere version for VMware components

Component	Minimum supported versions	Description
-----------	----------------------------	-------------

Component	Minimum supported versions	Description
Hypervisor	vSphere 8.0 Update 1 or later, or VMware Cloud Foundation 5.0 or later with virtual hardware version 15; VMware vSphere Foundation 9 or later, or VMware Cloud Foundation 9 or later	This hypervisor version is the minimum version that Red Hat Enterprise Linux CoreOS (RHCOS) supports. For more information about supported hardware on the latest version of Red Hat Enterprise Linux (RHEL) that is compatible with RHCOS, see Hardware on the Red Hat Customer Portal.
Networking (NSX)	vSphere 8.0 Update 1 or later, or VMware Cloud Foundation 5.0 or later; VMware vSphere Foundation 9 or later, or VMware Cloud Foundation 9 or later	Red Hat uses the Partner Certification process to verify NSX compatibility.
CPU micro-architecture	x86-64-v2 or higher	OpenShift Container Platform version 4.13 and later are based on the RHEL 9.2 host operating system, which raised the microarchitecture requirements to x86-64-v2. See Architectures in the RHEL documentation.



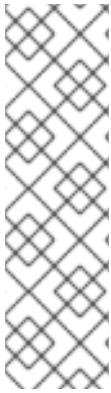
IMPORTANT

To ensure the best performance conditions for your cluster workloads that operate on Oracle® Cloud Infrastructure (OCI) and on the Oracle® Cloud VMware Solution (OCVS) service, ensure volume performance units (VPUs) for your block volume are sized for your workloads.

The following list provides some guidance in selecting the VPUs needed for specific performance needs:

- Test or proof of concept environment: 100 GB, and 20 to 30 VPUs.
- Base-production environment: 500 GB, and 60 VPUs.
- Heavy-use production environment: More than 500 GB, and 100 or more VPUs.

Consider allocating additional VPUs to give enough capacity for updates and scaling activities. See [Block Volume Performance Levels \(Oracle documentation\)](#).



NOTE

The following additional VMware vSphere Foundation and VMware Cloud Foundation components are outside the scope of Red Hat support:

- Management: VCF Operations, VCF Automation, VCF Fleet Management, and VCF Identity Broker.
- Networking: VMware NSX Container Plugin (NCP).
- Migration: VMware HCX.

3.1.2. VMware vSphere CSI Driver Operator requirements

To install the vSphere Container Storage Interface (CSI) Driver Operator, the following requirements must be met:

- VMware vSphere version 8.0 Update 1 or later; or VMware vSphere Foundation (VVF) 9; or VMware Cloud Foundation (VCF) 5 or later
- vCenter version 8.0 Update 1 or later; or VVF 9; or VCF 5 or later
- Virtual machines of hardware version 15 or later
- No third-party vSphere CSI driver already installed in the cluster

If a third-party vSphere CSI driver is present in the cluster, OpenShift Container Platform does not overwrite it. The presence of a third-party vSphere CSI driver prevents OpenShift Container Platform from updating to OpenShift Container Platform 4.13 or later.



NOTE

The VMware vSphere CSI Driver Operator is supported only on clusters deployed with **platform: vsphere** in the installation manifest.

You can create a custom role for the Container Storage Interface (CSI) driver, the vSphere CSI Driver Operator, and the vSphere Problem Detector Operator. The custom role can include privilege sets that assign a minimum set of permissions to each vSphere object. This means that the CSI driver, the vSphere CSI Driver Operator, and the vSphere Problem Detector Operator can establish a basic interaction with these objects.



IMPORTANT

Installing an OpenShift Container Platform cluster in a vCenter is tested against a full list of privileges as described in the "Required vCenter account privileges" section. By adhering to the full list of privileges, you can reduce the possibility of unexpected and unsupported behaviors that might occur when creating a custom role with a set of restricted privileges.

Additional resources

- To remove a third-party vSphere CSI driver, see [Removing a third-party vSphere CSI Driver](#).
- To update the hardware version for your vSphere nodes, see [Updating hardware on nodes running in vSphere](#).

- Minimum permissions for the storage components

3.1.3. Requirements for a cluster with user-provisioned infrastructure

For a cluster that contains user-provisioned infrastructure, you must deploy all of the required machines.

This section describes the requirements for deploying OpenShift Container Platform on user-provisioned infrastructure.

3.1.3.1. vCenter requirements

Before you install an OpenShift Container Platform cluster on your vCenter that uses infrastructure that you provided, you must prepare your environment.

3.1.3.1.1. Required vCenter account privileges

To install an OpenShift Container Platform cluster in a vCenter, your vSphere account must include privileges for reading and creating the required resources. Using an account that has global administrative privileges is the simplest way to access all of the necessary permissions.

Example 3.1. Roles and privileges required for installation in vSphere API

vSphere object for role	When required	Required privileges in vSphere API
vSphere vCenter	Always	Cns.Searchable InventoryService.Tagging.AttachTag InventoryService.Tagging.CreateCategory InventoryService.Tagging.CreateTag InventoryService.Tagging.DeleteCategory InventoryService.Tagging.DeleteTag InventoryService.Tagging.EditCategory InventoryService.Tagging.EditTag Sessions.ValidateSession StorageProfile.Update StorageProfile.View
vSphere vCenter Cluster	If VMs need creation in the cluster root	Host.Config.Storage Resource.AssignVMToPool VApp.AssignResourcePool VApp.Import VirtualMachine.Config.AddNewDisk

vSphere object for role	When required	Required privileges in vSphere API
vSphere vCenter Resource Pool	For a provided existing resource pool	Resource.AssignVMToPool VApp.AssignResourcePool VApp.Import VirtualMachine.Config.AddNewDisk
vSphere Datastore	Always	Datastore.AllocateSpace Datastore.Browse Datastore.FileManagement InventoryService.Tagging.ObjectAttachable
vSphere Port Group	Always	Network.Assign
Virtual Machine Folder	Always	InventoryService.Tagging.ObjectAttachable Resource.AssignVMToPool VApp.Import VirtualMachine.Config.AddExistingDisk VirtualMachine.Config.AddNewDisk VirtualMachine.Config.AddRemoveDevice VirtualMachine.Config.AdvancedConfig VirtualMachine.Config.Annotation VirtualMachine.Config.CPUCount VirtualMachine.Config.DiskExtend VirtualMachine.Config.DiskLease VirtualMachine.Config.EditDevice VirtualMachine.Config.Memory VirtualMachine.Config.RemoveDisk VirtualMachine.Config.Rename Host.Config.Storage VirtualMachine.Config.RestGuestInfo

vSphere object for role	When required	Required privileges in vSphere API
		VirtualMachine.Config.Resources VirtualMachine.Config.UpgradeVirtualHardware VirtualMachine.Interact.GuestControl VirtualMachine.Interact.PowerOff VirtualMachine.Interact.PowerOn VirtualMachine.Interact.Reset VirtualMachine.Inventory.Create VirtualMachine.Inventory.CreateFromExisting VirtualMachine.Inventory.Delete VirtualMachine.Provisioning.Clone VirtualMachine.Provisioning.MarkAsTemplate VirtualMachine.Provisioning.DeployTemplate
vSphere vCenter data center	VirtualMachine.Inventory.Create and VirtualMachine.Inventory.Delete privileges are optional if your cluster does not use the Machine API. See the "Minimum permissions for the Machine API" table.	InventoryService.Tagging.ObjectAttachable Resource.AssignVMTToPool VirtualMachine.Config.AddExistingDisk VirtualMachine.Config.AddNewDisk VirtualMachine.Config.AddRemoveDevice VirtualMachine.Config.AdvancedConfig VirtualMachine.Config.Annotation VirtualMachine.Config.CPUCount VirtualMachine.Config.DiskExtend VirtualMachine.Config.DiskLease VirtualMachine.Config.EditDevice VirtualMachine.Config.Memory VirtualMachine.Config.RemoveDisk

vSphere object for role	When required	Required privileges in vSphere API
		VirtualMachine.Config.Rename VirtualMachine.Config.Rese tGuestInfo VirtualMachine.Config.Reso urce VirtualMachine.Config.Setti ngs VirtualMachine.Config.Upgr adeVirtualHardware VirtualMachine.Interact.Gue stControl VirtualMachine.Interact.Pow erOff VirtualMachine.Interact.Pow erOn VirtualMachine.Interact.Res et VirtualMachine.Inventory.Cr eate VirtualMachine.Inventory.Cr eateFromExisting VirtualMachine.Inventory.D elete VirtualMachine.Provisionin g.Clone VirtualMachine.Provisionin g.DeployTemplate VirtualMachine.Provisionin g.MarkAsTemplate Folder.Create Folder.Delete

Example 3.2. Roles and privileges required for installation in vCenter graphical user interface (GUI)

vSphere object for role	When required	Required privileges in vCenter GUI

vSphere object for role	When required	Required privileges in vCenter GUI
vSphere vCenter	Always	Cns.Searchable "vSphere Tagging"."Assign or Unassign vSphere Tag" "vSphere Tagging"."Create vSphere Tag Category" "vSphere Tagging"."Create vSphere Tag" "vSphere Tagging"."Delete vSphere Tag Category" "vSphere Tagging"."Delete vSphere Tag" "vSphere Tagging"."Edit vSphere Tag Category" "vSphere Tagging"."Edit vSphere Tag" Sessions."Validate session" "VM storage policies"."Update VM storage policies" "VM storage policies"."View VM storage policies"
vSphere vCenter Cluster	For VMs creation in the cluster root	Host.Configuration."Storage partition configuration" Resource."Assign virtual machine to resource pool" VApp."Assign resource pool" VApp.Import "Virtual machine"."Change Configuration"."Add new disk"
vSphere vCenter Resource Pool	If providing an existing resource pool	Host.Configuration."Storage partition configuration" Resource."Assign virtual machine to resource pool" VApp."Assign resource pool" VApp.Import "Virtual machine"."Change Configuration"."Add new disk"
vSphere Datastore	Always	Datastore."Allocate space" Datastore."Browse datastore" Datastore."Low level file operations" "vSphere Tagging"."Assign or Unassign vSphere Tag on Object"

vSphere object for role	When required	Required privileges in vCenter GUI
vSphere Port Group	Always	Network ." Assign network "
Virtual Machine Folder	Always	"vSphere Tagging". "Assign or Unassign vSphere Tag on Object" Resource ." Assign virtual machine to resource pool " VApp.Import "Virtual machine". "Change Configuration". "Add existing disk" "Virtual machine". "Change Configuration". "Add new disk" "Virtual machine". "Change Configuration". "Add or remove device" "Virtual machine". "Change Configuration". "Advanced configuration" "Virtual machine". "Change Configuration". "Set annotation" "Virtual machine". "Change Configuration". "Change CPU count" "Virtual machine". "Change Configuration". "Extend virtual disk" "Virtual machine". "Change Configuration". "Acquire disk lease" "Virtual machine". "Change Configuration". "Modify device settings" "Virtual machine". "Change Configuration". "Change Memory" "Virtual machine". "Change Configuration". "Remove disk" "Virtual machine". "Change Configuration". "Rename" "Virtual machine". "Change Configuration". "Reset guest information" "Virtual machine". "Change Configuration". "Change resource" "Virtual machine". "Change Configuration". "Change

vSphere object for role	When required	Settings" Required privileges in vCenter GUI Virtual machine".Change Configuration".Upgrade virtual machine compatibility" "Virtual machine".Interaction."Guest operating system management by VIX API" "Virtual machine".Interaction."Power off" "Virtual machine".Interaction."Power on" "Virtual machine".Interaction.Reset "Virtual machine".Edit Inventory".Create new" "Virtual machine".Edit Inventory".Create from existing" "Virtual machine".Edit Inventory".Remove" "Virtual machine".Provisioning."Clone virtual machine" "Virtual machine".Provisioning."Mark as template" "Virtual machine".Provisioning."Deploy template"
vSphere vCenter data center	VirtualMachine.Inventory.Create and VirtualMachine.Inventory.Delete privileges are optional if your cluster does not use the Machine API.	"vSphere Tagging".Assign or Unassign vSphere Tag on Object" Resource .Assign virtual machine to resource pool" VApp.Import "Virtual machine".Change Configuration".Add existing disk" "Virtual machine".Change Configuration".Add new disk" "Virtual machine".Change Configuration".Add or remove device" "Virtual machine".Change Configuration".Advanced configuration" "Virtual machine".Change Configuration".Set annotation" "Virtual machine".Change

vSphere object for role	When required	Required privileges in vCenter GUI
		<p>"Configuration"."Change CPU count"</p> <p>"Virtual machine"."Change Configuration"</p> <p>"Virtual machine"."Extend virtual disk"</p> <p>"Virtual machine"."Change Configuration"."Acquire disk lease"</p> <p>"Virtual machine"."Change Configuration"."Modify device settings"</p> <p>"Virtual machine"."Change Configuration"."Change Memory"</p> <p>"Virtual machine"."Change Configuration"."Remove disk"</p> <p>"Virtual machine"."Change Configuration".Rename</p> <p>"Virtual machine"."Change Configuration"."Reset guest information"</p> <p>"Virtual machine"."Change Configuration"."Change resource"</p> <p>"Virtual machine"."Change Configuration"."Change Settings"</p> <p>"Virtual machine"."Change Configuration"."Upgrade virtual machine compatibility"</p> <p>"Virtual machine".Interaction."Guest operating system management by VIX API"</p> <p>"Virtual machine".Interaction."Power off"</p> <p>"Virtual machine".Interaction."Power on"</p> <p>"Virtual machine".Interaction.Reset</p> <p>"Virtual machine"."Edit Inventory"."Create new"</p> <p>"Virtual machine"."Edit Inventory"."Create from existing"</p> <p>"Virtual machine"."Edit Inventory"."Remove"</p> <p>"Virtual machine".Provisioning."Clone virtual machine"</p> <p>"Virtual</p>

vSphere object for role	When required	machine".Provisioning."Deploy template "Virtual machine".Provisioning."Mark as template" Folder."Create folder" Folder."Delete folder"
-------------------------	---------------	---

Additionally, the user requires some **ReadOnly** permissions, and some of the roles require permission to propagate the permissions to child objects. These settings vary depending on whether or not you install the cluster into an existing folder.

Example 3.3. Required permissions and propagation settings

vSphere object	When required	Propagate to children	Permissions required
vSphere vCenter	Always	False	Listed required privileges
vSphere vCenter data center	Existing folder	False	ReadOnly permission
vSphere vCenter Cluster	Existing resource pool	False	ReadOnly permission
	VMs in cluster root	True	Listed required privileges
vSphere vCenter Datastore	Always	False	Listed required privileges
vSphere Switch	Always	False	ReadOnly permission
vSphere Port Group	Always	False	Listed required privileges
vSphere vCenter Virtual Machine Folder	Existing folder	True	Listed required privileges
vSphere vCenter Resource Pool	Existing resource pool	True	Listed required privileges

For more information about creating an account with only the required privileges, see [vSphere Permissions and User Management Tasks](#) in the vSphere documentation.

3.1.3.1.2. Minimum required vCenter account privileges

After you create a custom role and assign privileges to the role, you can create permissions by selecting specific vSphere objects. You can then assign the custom role to a user or group for each object.

Before you create permissions or request for the creation of permissions for a vSphere object, decide what minimum permissions apply to the vSphere object. By doing this task, you can ensure a basic interaction exists between a vSphere object and OpenShift Container Platform architecture.



IMPORTANT

If you create a custom role and you do not assign privileges to it, the vSphere Server by default assigns a **Read Only** role to the custom role. Note that for the cloud provider API, the custom role only needs to inherit the privileges of the **Read Only** role.

Consider creating a custom role when an account with global administrative privileges does not meet your needs.



IMPORTANT

Red Hat does not support configuring an account without including the required privileges. Installing an OpenShift Container Platform cluster in a vCenter is tested against a full list of privileges as described in the "Required vCenter account privileges" section. By adhering to the full list of privileges, you can reduce the possibility of unexpected behaviors that might occur when creating a custom role with a restricted set of privileges.

The following tables specify how the required vCenter account privileges provided earlier in this document are relevant to different aspects of OpenShift Container Platform architecture.

Example 3.4. Minimum permissions for postinstallation management of components

vSphere object for role	When required	Required privileges
vSphere vCenter	Always	Cns.Searchable InventoryService.Tagging.AttachTag InventoryService.Tagging.CreateCategory InventoryService.Tagging.CreateTag InventoryService.Tagging.DeleteCategory InventoryService.Tagging.DeleteTag InventoryService.Tagging.EditCategory InventoryService.Tagging.EditTag Sessions.ValidateSession StorageProfile.Update StorageProfile.View

vSphere object for role	When required	Required privileges
vSphere vCenter Cluster	If you intend to create VMs in the cluster root	Host.Config.Storage Resource.AssignVMT oPool
vSphere vCenter Resource Pool	If you included an existing resource pool in the install-config.yaml file	Host.Config.Storage
vSphere Datastore	Always	Datastore.AllocateSpa ce Datastore.Browse Datastore.FileManage ment InventoryService.Tag ging.ObjectAttachable
vSphere Port Group	Always	Network.Assign
Virtual Machine Folder	Always	VirtualMachine.Config .AddExistingDisk VirtualMachine.Config .AddRemoveDevice VirtualMachine.Config .AdvancedConfig VirtualMachine.Config .Annotation VirtualMachine.Config .CPUCount VirtualMachine.Config .DiskExtend VirtualMachine.Config .Memory VirtualMachine.Config .Settings VirtualMachine.Interac t.PowerOff VirtualMachine.Interac t.PowerOn VirtualMachine.Invent ory.CreateFromExisti ng VirtualMachine.Invent ory.Delete VirtualMachine.Provisi oning.Clone VirtualMachine.Provisi oning.DeployTemplat e

vSphere object for role	When required	Required privileges
vSphere vCenter data center	<p>VirtualMachine.Inventory.Create and VirtualMachine.Inventory.Delete privileges are optional if your cluster does not use the Machine API. If your cluster does use the Machine API and you want to set the minimum set of permissions for the API, see the "Minimum permissions for the Machine API" table.</p>	Resource.AssignVMO ResourcePool VirtualMachine.Config .AddExistingDisk VirtualMachine.Config .AddRemoveDevice VirtualMachine.Interact .PowerOff VirtualMachine.Interact .PowerOn VirtualMachine.Provisioning .DeployTemplate e

Example 3.5. Minimum permissions for the storage components

vSphere object for role	When required	Required privileges
vSphere vCenter	Always	Cns.Searchable InventoryService.Tagging.CreateCategory InventoryService.Tagging.CreateTag InventoryService.Tagging.EditCategory InventoryService.Tagging.EditTag StorageProfile.Update StorageProfile.View
vSphere vCenter Cluster	If you intend to create VMs in the cluster root	Host.Config.Storage
vSphere vCenter Resource Pool	If you included an existing resource pool in the <code>install-config.yaml</code> file	Host.Config.Storage
vSphere Datastore	Always	Datastore.Browse Datastore.FileManagement InventoryService.Tagging.ObjectAttachable
vSphere Port Group	Always	Read Only

vSphere object for role	When required	Required privileges
Virtual Machine Folder	Always	VirtualMachine.Config .AddExistingDisk VirtualMachine.Config .AddRemoveDevice
vSphere vCenter data center	VirtualMachine.Inventory.Create and VirtualMachine.Inventory.Delete privileges are optional if your cluster does not use the Machine API. If your cluster does use the Machine API and you want to set the minimum set of permissions for the API, see the "Minimum permissions for the Machine API" table.	VirtualMachine.Config .AddExistingDisk VirtualMachine.Config .AddRemoveDevice

Example 3.6. Minimum permissions for the Machine API

vSphere object for role	When required	Required privileges
vSphere vCenter	Always	InventoryService.Tagging.AttachTag InventoryService.Tagging.CreateCategory InventoryService.Tagging.CreateTag InventoryService.Tagging.DeleteCategory InventoryService.Tagging.DeleteTag InventoryService.Tagging.EditCategory InventoryService.Tagging.EditTag Sessions.ValidateSession StorageProfile.Update StorageProfile.View
vSphere vCenter Cluster	If you intend to create VMs in the cluster root	Resource.AssignVMOPool
vSphere vCenter Resource Pool	If you included an existing resource pool in the install-config.yaml file	Read Only

vSphere object for role	When required	Required privileges
vSphere Datastore	Always	Datastore.AllocateSpace Datastore.Browse
vSphere Port Group	Always	Network.Assign
Virtual Machine Folder	Always	VirtualMachine.Config.AddRemoveDevice VirtualMachine.Config.AdvancedConfig VirtualMachine.Config.Annotation VirtualMachine.Config.CPUCount VirtualMachine.Config.DiskExtend VirtualMachine.Config.Memory VirtualMachine.Config.Settings VirtualMachine.Interact.PowerOff VirtualMachine.Interact.PowerOn VirtualMachine.Inventory.CreateFromExisting VirtualMachine.Inventory.Delete VirtualMachine.Provisioning.Clone VirtualMachine.Provisioning.DeployTemplate
vSphere vCenter data center	VirtualMachine.Inventory.Create and VirtualMachine.Inventory.Delete privileges are optional if your cluster does not use the Machine API.	Resource.AssignVMTpool VirtualMachine.Interact.PowerOff VirtualMachine.Interact.PowerOn VirtualMachine.Provisioning.DeployTemplate

3.1.3.1.3. Using OpenShift Container Platform with vMotion

If you intend on using vMotion in your vSphere environment, consider the following before installing an OpenShift Container Platform cluster.

- Using Storage vMotion can cause issues and is not supported.
- Using VMware compute vMotion to migrate the workloads for both OpenShift Container Platform compute machines and control plane machines is generally supported, where *generally* implies that you meet all VMware best practices for vMotion.

To help ensure the uptime of your compute and control plane nodes, ensure that you follow the VMware best practices for vMotion, and use VMware anti-affinity rules to improve the availability of OpenShift Container Platform during maintenance or hardware issues.

For more information about vMotion and anti-affinity rules, see the VMware vSphere documentation for [vMotion networking requirements](#) and [VM anti-affinity rules](#).

- If you are using VMware vSphere volumes in your pods, migrating a VM across datastores, either manually or through Storage vMotion, causes invalid references within OpenShift Container Platform persistent volume (PV) objects that can result in data loss.
- OpenShift Container Platform does not support selective migration of virtual machine disks (VMDKs) across datastores, using datastore clusters for VM provisioning or for dynamic or static provisioning of PVs, or using a datastore that is part of a datastore cluster for dynamic or static provisioning of PVs.



IMPORTANT

You can specify the path of any datastore that exists in a datastore cluster. By default, Storage Distributed Resource Scheduler (SDRS), which uses Storage vMotion, is automatically enabled for a datastore cluster. Red Hat does not support Storage vMotion, so you must disable SDRS to avoid data loss issues for your OpenShift Container Platform cluster.

If you must specify VMs across many datastores, use a **datastore** object to specify a failure domain in your cluster's **install-config.yaml** configuration file. For more information, see "VMware vSphere region and zone enablement".

3.1.3.1.4. Cluster resources

When you deploy an OpenShift Container Platform cluster that uses infrastructure that you provided, you must create the following resources in your vCenter instance:

- 1 Folder
- 1 Tag category
- 1 Tag
- Virtual machines:
 - 1 template
 - 1 temporary bootstrap node
 - 3 control plane nodes
 - 3 compute machines

Although these resources use 856 GB of storage, the bootstrap node gets deleted during the cluster installation process. At a minimum, a standard cluster requires 800 GB of storage.

If you deploy more compute machines, the OpenShift Container Platform cluster will use more storage.

3.1.3.1.5. Cluster limits

Available resources vary between clusters. A limit exists for the number of possible clusters within vCenter, primarily by available storage space and any limitations on the number of required resources. Be sure to consider both limitations to the vCenter resources that the cluster creates and the resources that you require to deploy a cluster, such as IP addresses and networks.

3.1.3.1.6. Networking requirements

You can use Dynamic Host Configuration Protocol (DHCP) for the network and configure the DHCP server to set persistent IP addresses to machines in your cluster. In the DHCP lease, you must configure the DHCP to use the default gateway.



NOTE

You do not need to use the DHCP for the network if you want to provision nodes with static IP addresses.

If you specify nodes or groups of nodes on different VLANs for a cluster that you want to install on user-provisioned infrastructure, you must ensure that machines in your cluster meet the requirements outlined in the "Network connectivity requirements" section of the *Networking requirements for user-provisioned infrastructure* document.

If you are installing to a restricted environment, the VM in your restricted network must have access to vCenter so that it can provision and manage nodes, persistent volume claims (PVCs), and other resources.



NOTE

Ensure that each OpenShift Container Platform node in the cluster has access to a Network Time Protocol (NTP) server that is discoverable by DHCP. Installation is possible without an NTP server. However, asynchronous server clocks can cause errors, which the NTP server prevents.

Additionally, you must create the following networking resources before you install the OpenShift Container Platform cluster:

3.1.3.1.6.1. DNS records

You must create DNS records for two static IP addresses in the appropriate DNS server for the vCenter instance that hosts your OpenShift Container Platform cluster. In each record, **<cluster_name>** is the cluster name and **<base_domain>** is the cluster base domain that you specify when you install the cluster. A complete DNS record takes the form: **<component>.<cluster_name>.<base_domain>..**

Table 3.3. Required DNS records

Component	Record	Description
API VIP	<code>api.<cluster_name>.<base_domain>.</code>	This DNS A/AAAA or CNAME (Canonical Name) record must point to the load balancer for the control plane machines. This record must be resolvable by both clients external to the cluster and from all the nodes within the cluster.
Ingress VIP	<code>*.apps.<cluster_name>.<base_domain>.</code>	A wildcard DNS A/AAAA or CNAME record that points to the load balancer that targets the machines that run the Ingress router pods, which are the worker nodes by default. This record must be resolvable by both clients external to the cluster and from all the nodes within the cluster.

Additional resources

- [Creating a compute machine set on vSphere](#)

3.1.3.2. Required machines for cluster installation

You must specify the minimum required machines or hosts for your cluster so that your cluster remains stable if a node fails.

The smallest OpenShift Container Platform clusters require the following hosts:



IMPORTANT

For a cluster that contains user-provisioned infrastructure, you must deploy all of the required machines.

Table 3.4. Minimum required hosts

Hosts	Description
One temporary bootstrap machine	The cluster requires the bootstrap machine to deploy the OpenShift Container Platform cluster on the three control plane machines. You can remove the bootstrap machine after you install the cluster.

Hosts	Description
Three control plane machines	The control plane machines run the Kubernetes and OpenShift Container Platform services that form the control plane.
At least two compute machines, which are also known as worker machines.	The workloads requested by OpenShift Container Platform users run on the compute machines.



IMPORTANT

To maintain high availability of your cluster, use separate physical hosts for these cluster machines.

The bootstrap and control plane machines must use Red Hat Enterprise Linux CoreOS (RHCOS) as the operating system. However, the compute machines can choose between Red Hat Enterprise Linux CoreOS (RHCOS), Red Hat Enterprise Linux (RHEL) 8.6 and later.

Note that RHCOS is based on Red Hat Enterprise Linux (RHEL) 9.2 and inherits all of its hardware certifications and requirements. See [Red Hat Enterprise Linux technology capabilities and limits](#) .

3.1.3.3. Minimum resource requirements for cluster installation

Each created cluster must meet minimum requirements so that the cluster runs as expected.

Table 3.5. Minimum resource requirements

Machine	Operating System	vCPU	Virtual RAM	Storage	Input/Output Per Second (IOPS) ^[1]
Bootstrap	RHCOS	4	16 GB	100 GB	300
Control plane	RHCOS	4	16 GB	100 GB	300
Compute	RHCOS	2	8 GB	100 GB	300

1. OpenShift Container Platform and Kubernetes are sensitive to disk performance, and faster storage is recommended, particularly for etcd on the control plane nodes which require a 10 ms p99 fsync duration. Note that on many cloud platforms, storage size and IOPS scale together, so you might need to over-allocate storage volume to obtain sufficient performance.
2. As with all user-provisioned installations, if you choose to use RHEL compute machines in your cluster, you take responsibility for all operating system life cycle management and maintenance, including performing system updates, applying patches, and completing all other required tasks. Use of RHEL 7 compute machines is deprecated and has been removed in OpenShift Container Platform 4.10 and later.



NOTE

For OpenShift Container Platform version 4.19, RHCOS is based on RHEL version 9.6, which updates the micro-architecture requirements. The following list contains the minimum instruction set architectures (ISA) that each architecture requires:

- x86-64 architecture requires x86-64-v2 ISA
- ARM64 architecture requires ARMv8.0-A ISA
- IBM Power architecture requires Power 9 ISA
- s390x architecture requires z14 ISA

For more information, see [Architectures](#) (RHEL documentation).

If an instance type for your platform meets the minimum requirements for cluster machines, it is supported to use in OpenShift Container Platform.



IMPORTANT

Do not use memory ballooning in OpenShift Container Platform clusters. Memory ballooning can cause cluster-wide instabilities, service degradation, or other undefined behaviors.

- Control plane machines should have committed memory equal to or greater than the published minimum resource requirements for a cluster installation.
- Compute machines should have a minimum reservation equal to or greater than the published minimum resource requirements for a cluster installation.

These minimum CPU and memory requirements do not account for resources required by user workloads.

For more information, see the Red Hat Knowledgebase article [Memory Ballooning and OpenShift](#).

Additional resources

- [Optimizing storage](#)

3.1.3.4. Requirements for encrypting virtual machines

You can encrypt your virtual machines prior to installing OpenShift Container Platform 4.19 by meeting the following requirements.

- You have configured a Standard key provider in vSphere. For more information, see [Adding a KMS to vCenter Server](#).



IMPORTANT

The Native key provider in vCenter is not supported. For more information, see [vSphere Native Key Provider Overview](#).

- You have enabled host encryption mode on all of the ESXi hosts that are hosting the cluster. For more information, see [Enabling host encryption mode](#).
- You have a vSphere account which has all cryptographic privileges enabled. For more information, see [Cryptographic Operations Privileges](#).

When you deploy the OVF template in the section titled "Installing RHCOS and starting the OpenShift Container Platform bootstrap process", select the option to "Encrypt this virtual machine" when you are selecting storage for the OVF template. After completing cluster installation, create a storage class that uses the encryption storage policy you used to encrypt the virtual machines.

Additional resources

- [Creating an encrypted storage class](#)

3.1.3.5. Certificate signing requests management

On user-provisioned infrastructure, you must provide a mechanism for approving cluster certificate signing requests (CSRs) after installation when your cluster has limited access to automatic machine management.

The **kube-controller-manager** only approves the kubelet client CSRs. The **machine-approver** cannot guarantee the validity of a serving certificate that is requested by using kubelet credentials because it cannot confirm that the correct machine issued the request. You must determine and implement a method of verifying the validity of the kubelet serving certificate requests and approving them.

3.1.3.6. Networking requirements for user-provisioned infrastructure

You must configure networking for all the Red Hat Enterprise Linux CoreOS (RHCOS) machines in **initramfs** during boot, so that they can fetch their Ignition config files.



IMPORTANT

Ensure you enable the **disk.EnableUUID** parameter on all virtual machines in your cluster.

During the initial boot, the machines require an IP address configuration that is set either through a DHCP server or statically by providing the required boot options. After a network connection is established, the machines download their Ignition config files from an HTTP or HTTPS server. The Ignition config files are then used to set the exact state of each machine. The Machine Config Operator completes more changes to the machines, such as the application of new certificates or keys, after installation.



NOTE

- Consider using a DHCP server for long-term management of the cluster machines. Ensure that the DHCP server is configured to provide persistent IP addresses, DNS server information, and hostnames to the cluster machines.
- If a DHCP service is not available for your user-provisioned infrastructure, you can instead provide the IP networking configuration and the address of the DNS server to the nodes at RHCOS install time. These can be passed as boot arguments if you are installing from an ISO image. See the *Installing RHCOS and starting the OpenShift Container Platform bootstrap process* section for more information about static IP provisioning and advanced networking options.

The Kubernetes API server must be able to resolve the node names of the cluster machines. If the API servers and worker nodes are in different zones, you can configure a default DNS search zone to allow the API server to resolve the node names. Another supported approach is to always refer to hosts by their fully-qualified domain names in both the node objects and all DNS requests.

3.1.3.6.1. Setting the cluster node hostnames through DHCP

On Red Hat Enterprise Linux CoreOS (RHCOS) machines, the hostname is set through NetworkManager. By default, the machines obtain their hostname through DHCP. If the hostname is not provided by DHCP, set statically through kernel arguments, or another method, it is obtained through a reverse DNS lookup. Reverse DNS lookup occurs after the network has been initialized on a node and can take time to resolve. Other system services can start prior to this and detect the hostname as **localhost** or similar. You can avoid this by using DHCP to provide the hostname for each cluster node.

Additionally, setting the hostnames through DHCP can bypass any manual DNS record name configuration errors in environments that have a DNS split-horizon implementation.

3.1.3.6.2. Network connectivity requirements

You must configure the network connectivity between machines to allow OpenShift Container Platform cluster components to communicate. Each machine must be able to resolve the hostnames of all other machines in the cluster.

This section provides details about the ports that are required.



IMPORTANT

In connected OpenShift Container Platform environments, all nodes are required to have internet access to pull images for platform containers and provide telemetry data to Red Hat.

Table 3.6. Ports used for all-machine to all-machine communications

Protocol	Port	Description
ICMP	N/A	Network reachability tests
TCP	1936	Metrics
	9000-9999	Host level services, including the node exporter on ports 9100-9101 and the Cluster Version Operator on port 9099 .
	10250-10259	The default ports that Kubernetes reserves
	22623	The port handles traffic from the Machine Config Server and directs the traffic to the control plane machines.
UDP	6081	Geneve
	9000-9999	Host level services, including the node exporter on ports 9100-9101 .

Protocol	Port	Description
	500	IPsec IKE packets
	4500	IPsec NAT-T packets
	123	Network Time Protocol (NTP) on UDP port 123 . If an external NTP time server is configured, you must open UDP port 123 .
	TCP/UDP	30000-32767
Kubernetes node port	ESP	N/A

Table 3.7. Ports used for all-machine to control plane communications

Protocol	Port	Description
TCP	6443	Kubernetes API

Table 3.8. Ports used for control plane machine to control plane machine communications

Protocol	Port	Description
TCP	2379-2380	etcd server and peer ports

3.1.3.6.3. NTP configuration for user-provisioned infrastructure

OpenShift Container Platform clusters are configured to use a public Network Time Protocol (NTP) server by default. If you want to use a local enterprise NTP server, or if your cluster is being deployed in a disconnected network, you can configure the cluster to use a specific time server. For more information, see the documentation for *Configuring chrony time service* .

If a DHCP server provides NTP server information, the chrony time service on the Red Hat Enterprise Linux CoreOS (RHCOS) machines read the information and can sync the clock with the NTP servers.

Additional resources

- [Configuring chrony time service](#)

3.1.3.7. User-provisioned DNS requirements

In OpenShift Container Platform deployments, you must ensure that cluster components meet certain DNS name resolution criteria for internal communication, certificate validation, and automated node discovery purposes.

The following is a list of required cluster components:

- The Kubernetes API
- The OpenShift Container Platform application wildcard
- The bootstrap, control plane, and compute machines

Reverse DNS resolution is also required for the Kubernetes API, the bootstrap machine, the control plane machines, and the compute machines.

DNS A/AAAA or CNAME records are used for name resolution and PTR records are used for reverse name resolution. The reverse records are important because Red Hat Enterprise Linux CoreOS (RHCOS) uses the reverse records to set the hostnames for all the nodes, unless the hostnames are provided by DHCP. Additionally, the reverse records are used to generate the certificate signing requests (CSR) that OpenShift Container Platform needs to operate.



NOTE

It is recommended to use a DHCP server to provide the hostnames to each cluster node. See the *DHCP recommendations for user-provisioned infrastructure* section for more information.

The following DNS records are required for a user-provisioned OpenShift Container Platform cluster and they must be in place before installation. In each record, `<cluster_name>` is the cluster name and `<base_domain>` is the base domain that you specify in the `install-config.yaml` file. A complete DNS record takes the form: `<component>.<cluster_name>.<base_domain>..`

Table 3.9. Required DNS records

Component	Record	Description
Kubernetes API	<code>api.<cluster_name>.<base_domain>.</code>	A DNS A/AAAA or CNAME record, and a DNS PTR record, to identify the API load balancer. These records must be resolvable by both clients external to the cluster and from all the nodes within the cluster.
	<code>api-int.<cluster_name>.<base_domain>.</code>	A DNS A/AAAA or CNAME record, and a DNS PTR record, to internally identify the API load balancer. These records must be resolvable from all the nodes within the cluster.



IMPORTANT

The API server must be able to resolve the worker nodes by the hostnames that are recorded in Kubernetes. If the API server cannot resolve the node names, then proxied API calls can fail, and you cannot retrieve logs from pods.

Component	Record	Description
Routes	<code>*.apps.<cluster_name>.<base_domain>.</code>	A wildcard DNS A/AAAA or CNAME record that refers to the application ingress load balancer. The application ingress load balancer targets the machines that run the Ingress Controller pods. The Ingress Controller pods run on the compute machines by default. These records must be resolvable by both clients external to the cluster and from all the nodes within the cluster. For example, <code>console-openshift-console.apps.<cluster_name>.<base_domain></code> is used as a wildcard route to the OpenShift Container Platform console.
Bootstrap machine	<code>bootstrap.<cluster_name>.<base_domain>.</code>	A DNS A/AAAA or CNAME record, and a DNS PTR record, to identify the bootstrap machine. These records must be resolvable by the nodes within the cluster.
Control plane machines	<code><control_plane><n>.<cluster_name>.<base_domain>.</code>	DNS A/AAAA or CNAME records and DNS PTR records to identify each machine for the control plane nodes. These records must be resolvable by the nodes within the cluster.
Compute machines	<code><compute><n>.<cluster_name>.<base_domain>.</code>	DNS A/AAAA or CNAME records and DNS PTR records to identify each machine for the worker nodes. These records must be resolvable by the nodes within the cluster.



NOTE

In OpenShift Container Platform 4.4 and later, you do not need to specify etcd host and SRV records in your DNS configuration.

TIP

You can use the `dig` command to verify name and reverse name resolution. See the section on [Validating DNS resolution for user-provisioned infrastructure](#) for detailed validation steps.

3.1.3.7.1. Example DNS configuration for user-provisioned clusters

Reference the example DNS configurations to understand how A and PTR record configuration samples meet the DNS requirements for deploying OpenShift Container Platform on user-provisioned infrastructure.

The DNS configuration examples provided here are for reference only and are not meant to provide advice for choosing one DNS solution over another.

In the examples, the cluster name is `ocp4` and the base domain is `example.com`.

The following example is a BIND zone file that shows sample DNS A records for name resolution in a user-provisioned cluster.



NOTE

In the example, the same load balancer is used for the Kubernetes API and application ingress traffic. In production scenarios, you can deploy the API and application ingress load balancers separately so that you can scale the load balancer infrastructure for each in isolation.

```
$TTL 1W
@ IN SOA ns1.example.com. root (
    2019070700 ; serial
    3H ; refresh (3 hours)
    30M ; retry (30 minutes)
    2W ; expiry (2 weeks)
    1W ) ; minimum (1 week)
IN NS ns1.example.com.
IN MX 10 smtp.example.com.
;
;
ns1.example.com. IN A 192.168.1.5
smtp.example.com. IN A 192.168.1.5
;
helper.example.com. IN A 192.168.1.5
helper.ocp4.example.com. IN A 192.168.1.5
;
api.ocp4.example.com. IN A 192.168.1.5
api-int.ocp4.example.com. IN A 192.168.1.5
;
*.apps.ocp4.example.com. IN A 192.168.1.5
;
bootstrap.ocp4.example.com. IN A 192.168.1.96
;
control-plane0.ocp4.example.com. IN A 192.168.1.97
control-plane1.ocp4.example.com. IN A 192.168.1.98
;
control-plane2.ocp4.example.com. IN A 192.168.1.99
;
compute0.ocp4.example.com. IN A 192.168.1.11
compute1.ocp4.example.com. IN A 192.168.1.7
;
:EOF
```

where:

api.ocp4.example.com.

Provides name resolution for the Kubernetes API. The record refers to the IP address of the API load balancer.

api-int.ocp4.example.com.

Provides name resolution for the Kubernetes API. The record refers to the IP address of the API load balancer and is used for internal cluster communications.

***.apps.ocp4.example.com.**

Provides name resolution for the wildcard routes. The record refers to the IP address of the application ingress load balancer. The application ingress load balancer targets the machines that run the Ingress Controller pods.

bootstrap.ocp4.example.com

Provides name resolution for the bootstrap machine.

control-plane0.ocp4.example.com

Provides name resolution for the control plane machines.

compute0.ocp4.example.com.

Provides name resolution for the compute machines.

The following example BIND zone file shows sample PTR records for reverse name resolution in a user-provisioned cluster:

```
$TTL 1W
@ IN SOA ns1.example.com. root (
    2019070700 ; serial
    3H ; refresh (3 hours)
    30M ; retry (30 minutes)
    2W ; expiry (2 weeks)
    1W ) ; minimum (1 week)
IN NS ns1.example.com.

;
5.1.168.192.in-addr.arpa. IN PTR api.ocp4.example.com.
5.1.168.192.in-addr.arpa. IN PTR api-int.ocp4.example.com.
;
96.1.168.192.in-addr.arpa. IN PTR bootstrap.ocp4.example.com.
;
97.1.168.192.in-addr.arpa. IN PTR control-plane0.ocp4.example.com.
98.1.168.192.in-addr.arpa. IN PTR control-plane1.ocp4.example.com.
;
99.1.168.192.in-addr.arpa. IN PTR control-plane2.ocp4.example.com.
;
11.1.168.192.in-addr.arpa. IN PTR compute0.ocp4.example.com.
7.1.168.192.in-addr.arpa. IN PTR compute1.ocp4.example.com.
;
:EOF
```

where:

api.ocp4.example.com.

Provides reverse DNS resolution for the Kubernetes API. The PTR record refers to the record name of the API load balancer.

api-int.ocp4.example.com.

Provides reverse DNS resolution for the Kubernetes API. The PTR record refers to the record name of the API load balancer and is used for internal cluster communications.

bootstrap.ocp4.example.com.

Provides reverse DNS resolution for the bootstrap machine.

control-plane0.ocp4.example.com.

Provides rebootstrap.ocp4.example.com.verse DNS resolution for the control plane machines.

compute0.ocp4.example.com.

Provides reverse DNS resolution for the compute machines.



NOTE

A PTR record is not required for the OpenShift Container Platform application wildcard.

3.1.3.8. Load balancing requirements for user-provisioned infrastructure

Before you install OpenShift Container Platform, you must provision the API and application Ingress load balancing infrastructure. In production scenarios, you can deploy the API and application Ingress load balancers separately so that you can scale the load balancer infrastructure for each in isolation.



NOTE

If you want to deploy the API and application Ingress load balancers with a Red Hat Enterprise Linux (RHEL) instance, you must purchase the RHEL subscription separately.

The load balancing infrastructure must meet the following requirements:

- API load balancer: Provides a common endpoint for users, both human and machine, to interact with and configure the platform. Configure the following conditions:
 - Layer 4 load balancing only. This can be referred to as Raw TCP or SSL Passthrough mode.
 - A stateless load balancing algorithm. The options vary based on the load balancer implementation.



IMPORTANT

Do not configure session persistence for an API load balancer. Configuring session persistence for a Kubernetes API server might cause performance issues from excess application traffic for your OpenShift Container Platform cluster and the Kubernetes API that runs inside the cluster.

Configure the following ports on both the front and back of the API load balancers:

Port	Back-end machines (pool members)	Internal	External	Description
6443	Bootstrap and control plane. You remove the bootstrap machine from the load balancer after the bootstrap machine initializes the cluster control plane. You must configure the /readyz endpoint for the API server health check probe.	X	X	Kubernetes API server
22623	Bootstrap and control plane. You remove the bootstrap machine from the load balancer after the bootstrap machine initializes the cluster control plane.	X		Machine config server



NOTE

The load balancer must be configured to take a maximum of 30 seconds from the time the API server turns off the `/readyz` endpoint to the removal of the API server instance from the pool. Within the time frame after `/readyz` returns an error or becomes healthy, the endpoint must have been removed or added. Probing every 5 or 10 seconds, with two successful requests to become healthy and three to become unhealthy, are well-tested values.

- Application Ingress load balancer: Provides an ingress point for application traffic flowing in from outside the cluster. A working configuration for the Ingress router is required for an OpenShift Container Platform cluster. Configure the following conditions:
 - Layer 4 load balancing only. This can be referred to as Raw TCP or SSL Passthrough mode.
 - A connection-based or session-based persistence is recommended, based on the options available and types of applications that will be hosted on the platform.

TIP

If the true IP address of the client can be seen by the application Ingress load balancer, enabling source IP-based session persistence can improve performance for applications that use end-to-end TLS encryption.

Configure the following ports on both the front and back of the load balancers:

Table 3.10. Application Ingress load balancer

Port	Back-end machines (pool members)	Internal	External	Description
443	The machines that run the Ingress Controller pods, compute, or worker, by default.	X	X	HTTPS traffic
80	The machines that run the Ingress Controller pods, compute, or worker, by default.	X	X	HTTP traffic



NOTE

If you are deploying a three-node cluster with zero compute nodes, the Ingress Controller pods run on the control plane nodes. In three-node cluster deployments, you must configure your application Ingress load balancer to route HTTP and HTTPS traffic to the control plane nodes.

3.1.3.8.1. Example load balancer configuration for user-provisioned clusters

Reference the example API and application Ingress load balancer configuration so that you can understand how to meet the load balancing requirements for user-provisioned clusters.

The sample is an `/etc/haproxy/haproxy.cfg` configuration for an HAProxy load balancer. The example is not meant to provide advice for choosing one load balancing solution over another.

TIP

If you are using HAProxy as a load balancer, you can check that the **haproxy** process is listening on ports **6443**, **22623**, **443**, and **80** by running **netstat -nltpu** on the HAProxy node.

In the example, the same load balancer is used for the Kubernetes API and application ingress traffic. In production scenarios, you can deploy the API and application ingress load balancers separately so that you can scale the load balancer infrastructure for each in isolation.

**NOTE**

If you are using HAProxy as a load balancer and SELinux is set to **enforcing**, you must ensure that the HAProxy service can bind to the configured TCP port by running **setsebool -P haproxy_connect_any=1**.

Sample API and application Ingress load balancer configuration

```
global
  log    127.0.0.1 local2
  pidfile /var/run/haproxy.pid
  maxconn 4000
  daemon
defaults
  mode      http
  log       global
  option    dontlognull
  option http-server-close
  option    redispatch
  retries   3
  timeout http-request 10s
  timeout queue    1m
  timeout connect   10s
  timeout client    1m
  timeout server    1m
  timeout http-keep-alive 10s
  timeout check     10s
  maxconn   3000
listen api-server-6443
  bind *:6443
  mode tcp
  option httpchk GET /readyz HTTP/1.0
  option log-health-checks
  balance roundrobin
  server bootstrap bootstrap.ocp4.example.com:6443 verify none check check-ssl inter 10s fall 2 rise
3 backup
  server master0 master0.ocp4.example.com:6443 weight 1 verify none check check-ssl inter 10s fall
2 rise 3
  server master1 master1.ocp4.example.com:6443 weight 1 verify none check check-ssl inter 10s fall
2 rise 3
  server master2 master2.ocp4.example.com:6443 weight 1 verify none check check-ssl inter 10s fall
2 rise 3
listen machine-config-server-22623
  bind *:22623
  mode tcp
  server bootstrap bootstrap.ocp4.example.com:22623 check inter 1s backup
```

```

server master0 master0.ocp4.example.com:22623 check inter 1s
server master1 master1.ocp4.example.com:22623 check inter 1s
server master2 master2.ocp4.example.com:22623 check inter 1s
listen ingress-router-443
  bind *:443
  mode tcp
  balance source
  server compute0 compute0.ocp4.example.com:443 check inter 1s
  server compute1 compute1.ocp4.example.com:443 check inter 1s
listen ingress-router-80
  bind *:80
  mode tcp
  balance source
  server compute0 compute0.ocp4.example.com:80 check inter 1s
  server compute1 compute1.ocp4.example.com:80 check inter 1s

```

where:

listen api-server-6443

Port **6443** handles the Kubernetes API traffic and points to the control plane machines.

server bootstrap bootstrap.ocp4.example.com

The bootstrap entries must be in place before the OpenShift Container Platform cluster installation and they must be removed after the bootstrap process is complete.

listen machine-config-server

Port **22623** handles the machine config server traffic and points to the control plane machines.

listen ingress-router-443

Port **443** handles the HTTPS traffic and points to the machines that run the Ingress Controller pods.

The Ingress Controller pods run on the compute machines by default.

listen ingress-router-80

Port **80** handles the HTTP traffic and points to the machines that run the Ingress Controller pods.

The Ingress Controller pods run on the compute machines by default.



NOTE

If you are deploying a three-node cluster with zero compute nodes, the Ingress Controller pods run on the control plane nodes. In three-node cluster deployments, you must configure your application Ingress load balancer to route HTTP and HTTPS traffic to the control plane nodes.

3.2. PREPARING TO INSTALL A CLUSTER USING USER-PROVISIONED INFRASTRUCTURE

You prepare to install an OpenShift Container Platform cluster on vSphere by completing the following steps:

- Downloading the installation program.

**NOTE**

If you are installing in a disconnected environment, you extract the installation program from the mirrored content. For more information, see [Mirroring images for a disconnected installation](#).

- Installing the OpenShift CLI (**oc**).

**NOTE**

If you are installing in a disconnected environment, install **oc** to the mirror host.

- Generating an SSH key pair. You can use this key pair to authenticate into the OpenShift Container Platform cluster's nodes after it is deployed.
- Preparing the user-provisioned infrastructure.
- Validating DNS resolution.

3.2.1. Obtaining the installation program

Before you install OpenShift Container Platform, download the installation file on the host you are using for installation.

Prerequisites

- You have a computer that runs Linux or macOS, with 500 MB of local disk space.

Procedure

1. Go to the [Cluster Type](#) page on the Red Hat Hybrid Cloud Console. If you have a Red Hat account, log in with your credentials. If you do not, create an account.

TIP

You can also [download the binaries for a specific OpenShift Container Platform release](#).

2. Select your infrastructure provider from the **Run it yourself** section of the page.
3. Select your host operating system and architecture from the dropdown menus under **OpenShift Installer** and click **Download Installer**.
4. Place the downloaded file in the directory where you want to 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 the installation program and the files that the installation program creates after you finish installing the cluster. Both of the files are required to delete the cluster.
- Deleting the files created by the installation program does not remove your cluster, even if the cluster failed during installation. To remove your cluster, complete the OpenShift Container Platform uninstallation procedures for your specific cloud provider.

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

```
$ tar -xvf openshift-install-linux.tar.gz
```

6. Download your installation [pull secret from Red Hat OpenShift Cluster Manager](#). 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.

TIP

Alternatively, you can retrieve the installation program from the [Red Hat Customer Portal](#), where you can specify a version of the installation program to download. However, you must have an active subscription to access this page.

3.2.2. Installing the OpenShift CLI on Linux

To manage your cluster and deploy applications from the command line, install the OpenShift CLI (**oc**) binary on Linux.



IMPORTANT

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

Download and install the new version of **oc**.

Procedure

1. Navigate to the [Download OpenShift Container Platform](#) page on the Red Hat Customer Portal.
2. Select the architecture from the **Product Variant** list.
3. Select the appropriate version from the **Version** list.
4. Click **Download Now** next to the **OpenShift v4.19 Linux Clients** entry and save the file.
5. Unpack the archive:

```
$ tar xvf <file>
```

6. Place the **oc** binary in a directory that is on your **PATH**.

To check your **PATH**, execute the following command:

```
$ echo $PATH
```

Verification

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

```
$ oc <command>
```

3.2.3. Installing the OpenShift CLI on Windows

To manage your cluster and deploy applications from the command line, install OpenShift CLI (**oc**) binary on Windows.



IMPORTANT

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

Download and install the new version of **oc**.

Procedure

1. Navigate to the [Download OpenShift Container Platform](#) page on the Red Hat Customer Portal.
2. Select the appropriate version from the **Version** list.
3. Click **Download Now** next to the **OpenShift v4.19 Windows Client** entry and save the file.
4. Extract the archive with a ZIP program.
5. Move the **oc** binary to a directory that is on your **PATH** variable.

To check your **PATH** variable, open the command prompt and execute the following command:

```
C:> path
```

Verification

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

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

3.2.4. Installing the OpenShift CLI on macOS

To manage your cluster and deploy applications from the command line, install the OpenShift CLI (**oc**) binary on macOS.



IMPORTANT

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

Download and install the new version of **oc**.

Procedure

1. Navigate to the [Download OpenShift Container Platform](#) page on the Red Hat Customer Portal.
2. Select the architecture from the **Product Variant** list.
3. Select the appropriate version from the **Version** list.
4. Click **Download Now** next to the **OpenShift v4.19 macOS Clients** entry and save the file.



NOTE

For macOS arm64, choose the **OpenShift v4.19 macOS arm64 Client** entry.

5. Unpack and unzip the archive.
6. Move the **oc** binary to a directory on your **PATH** variable.

To check your **PATH** variable, open a terminal and execute the following command:

```
$ echo $PATH
```

Verification

- Verify your installation by using an **oc** command:

```
$ oc <command>
```

3.2.5. Generating a key pair for cluster node SSH access

To enable secure, passwordless SSH access to your cluster nodes, provide an SSH public key during the OpenShift Container Platform installation. This ensures that the installation program automatically configures the Red Hat Enterprise Linux CoreOS (RHCOS) nodes for remote authentication through the **core** user.

The SSH public key gets added to the `~/.ssh/authorized_keys` list for the **core** user on each node. After the key is passed to the Red Hat Enterprise Linux CoreOS (RHCOS) nodes through their Ignition config files, you can use the key pair to SSH in to the RHCOS nodes as the user **core**. To access the nodes through SSH, the private key identity must be managed by SSH for your local user.

If you want to SSH in to your cluster nodes to perform installation debugging or disaster recovery, you must provide the SSH public key during the installation process. The `./openshift-install gather` command also requires the SSH public key to be in place on the cluster nodes.



IMPORTANT

Do not skip this procedure in production environments, where disaster recovery and debugging is required.



NOTE

You must use a local key, not one that you configured with platform-specific approaches.

Procedure

1. If you do not have an existing SSH key pair on your local machine to use for authentication onto your cluster nodes, create one. For example, on a computer that uses a Linux operating system, run the following command:

```
$ ssh-keygen -t ed25519 -N "" -f <path>/<file_name>
```

Specifies the path and file name, such as `~/.ssh/id_ed25519`, of the new SSH key. If you have an existing key pair, ensure your public key is in the your `~/.ssh` directory.



NOTE

If you plan to install an OpenShift Container Platform cluster that uses 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, do not create a key that uses the **ed25519** algorithm. Instead, create a key that uses the **rsa** or **ecdsa** algorithm.

2. View the public SSH key:

```
$ cat <path>/<file_name>.pub
```

For example, run the following to view the `~/.ssh/id_ed25519.pub` public key:

```
$ cat ~/.ssh/id_ed25519.pub
```

3. Add the SSH private key identity to the SSH agent for your local user, if it has not already been added. SSH agent management of the key is required for password-less SSH authentication onto your cluster nodes, or if you want to use the `./openshift-install gather` command.



NOTE

On some distributions, default SSH private key identities such as `~/.ssh/id_rsa` and `~/.ssh/id_dsa` are managed automatically.

- a. If the **ssh-agent** process is not already running for your local user, start it as a background task:

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

Example output



Agent pid 31874



NOTE

If your cluster is in FIPS mode, only use FIPS-compliant algorithms to generate the SSH key. The key must be either RSA or ECDSA.

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

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

Specifies the path and file name for your SSH private key, such as `~/.ssh/id_ed25519`

Example output

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

Next steps

- When you install OpenShift Container Platform, provide the SSH public key to the installation program. If you install a cluster on infrastructure that you provision, you must provide the key to the installation program.

3.2.6. Preparing the user-provisioned infrastructure

To ensure a successful deployment and meet cluster requirements in OpenShift Container Platform, prepare your user-provisioned infrastructure before starting the installation. Configuring your compute, network, and storage components in advance provides the stable foundation necessary for the installation program to function correctly.

This section provides details about the high-level steps required to set up your cluster infrastructure in preparation for an OpenShift Container Platform installation. This includes configuring IP networking and network connectivity for your cluster nodes, enabling the required ports through your firewall, and setting up the required DNS and load balancing infrastructure.

After preparation, your cluster infrastructure must meet the requirements outlined in the *Requirements for a cluster with user-provisioned infrastructure* section.

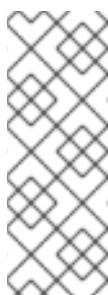
Prerequisites

- You have reviewed the [OpenShift Container Platform 4.x Tested Integrations](#) page.
- You have reviewed the infrastructure requirements detailed in the *Requirements for a cluster with user-provisioned infrastructure* section.

Procedure

- If you are using DHCP to provide the IP networking configuration to your cluster nodes, configure your DHCP service.
 - Add persistent IP addresses for the nodes to your DHCP server configuration. In your configuration, match the MAC address of the relevant network interface to the intended IP address for each node.

- b. When you use DHCP to configure IP addressing for the cluster machines, the machines also obtain the DNS server information through DHCP. Define the persistent DNS server address that is used by the cluster nodes through your DHCP server configuration.



NOTE

If you are not using a DHCP service, you must provide the IP networking configuration and the address of the DNS server to the nodes at RHCOS install time. These can be passed as boot arguments if you are installing from an ISO image. See the *Installing RHCOS and starting the OpenShift Container Platform bootstrap* process section for more information about static IP provisioning and advanced networking options.

- c. Define the hostnames of your cluster nodes in your DHCP server configuration. See the *Setting the cluster node hostnames through DHCP* section for details about hostname considerations.



NOTE

If you are not using a DHCP service, the cluster nodes obtain their hostname through a reverse DNS lookup.

2. Ensure that your network infrastructure provides the required network connectivity between the cluster components. See the *Networking requirements for user-provisioned infrastructure* section for details about the requirements.
3. Configure your firewall to enable the ports required for the OpenShift Container Platform cluster components to communicate. See *Networking requirements for user-provisioned infrastructure* section for details about the ports that are required.



IMPORTANT

By default, port **1936** is accessible for an OpenShift Container Platform cluster, because each control plane node needs access to this port.

Avoid using the Ingress load balancer to expose this port, because doing so might result in the exposure of sensitive information, such as statistics and metrics, related to Ingress Controllers.

4. Setup the required DNS infrastructure for your cluster.
 - a. Configure DNS name resolution for the Kubernetes API, the application wildcard, the bootstrap machine, the control plane machines, and the compute machines.
 - b. Configure reverse DNS resolution for the Kubernetes API, the bootstrap machine, the control plane machines, and the compute machines.
- See the *User-provisioned DNS requirements* section for more information about the OpenShift Container Platform DNS requirements.
5. Validate your DNS configuration.
 - a. From your installation node, run DNS lookups against the record names of the Kubernetes API, the wildcard routes, and the cluster nodes. Validate that the IP addresses in the responses correspond to the correct components.

- b. From your installation node, run reverse DNS lookups against the IP addresses of the load balancer and the cluster nodes. Validate that the record names in the responses correspond to the correct components.

See the *Validating DNS resolution for user-provisioned infrastructure* section for detailed DNS validation steps.

- 6. Provision the required API and application ingress load balancing infrastructure. See the *Load balancing requirements for user-provisioned infrastructure* section for more information about the requirements.



NOTE

Some load balancing solutions require the DNS name resolution for the cluster nodes to be in place before the load balancing is initialized.

3.2.6.1. Example load balancer configuration for user-provisioned clusters

Reference the example API and application Ingress load balancer configuration so that you can understand how to meet the load balancing requirements for user-provisioned clusters.

The sample is an **/etc/haproxy/haproxy.cfg** configuration for an HAProxy load balancer. The example is not meant to provide advice for choosing one load balancing solution over another.

TIP

If you are using HAProxy as a load balancer, you can check that the **haproxy** process is listening on ports **6443, 22623, 443, and 80** by running **netstat -nltpu** on the HAProxy node.

In the example, the same load balancer is used for the Kubernetes API and application ingress traffic. In production scenarios, you can deploy the API and application ingress load balancers separately so that you can scale the load balancer infrastructure for each in isolation.



NOTE

If you are using HAProxy as a load balancer and SELinux is set to **enforcing**, you must ensure that the HAProxy service can bind to the configured TCP port by running **setsebool -P haproxy_connect_any=1**.

Sample API and application Ingress load balancer configuration

```

global
  log    127.0.0.1 local2
  pidfile /var/run/haproxy.pid
  maxconn 4000
  daemon
defaults
  mode      http
  log       global
  option    dontlognull
  option http-server-close
  option    redispatch
  retries   3
  timeout http-request 10s

```

```

timeout queue      1m
timeout connect    10s
timeout client      1m
timeout server      1m
timeout http-keep-alive 10s
timeout check      10s
maxconn          3000
listen api-server-6443
  bind *:6443
  mode tcp
  option httpchk GET /readyz HTTP/1.0
  option log-health-checks
  balance roundrobin
  server bootstrap bootstrap.ocp4.example.com:6443 verify none check check-ssl inter 10s fall 2 rise
3 backup
  server master0 master0.ocp4.example.com:6443 weight 1 verify none check check-ssl inter 10s fall
2 rise 3
  server master1 master1.ocp4.example.com:6443 weight 1 verify none check check-ssl inter 10s fall
2 rise 3
  server master2 master2.ocp4.example.com:6443 weight 1 verify none check check-ssl inter 10s fall
2 rise 3
listen machine-config-server-22623
  bind *:22623
  mode tcp
  server bootstrap bootstrap.ocp4.example.com:22623 check inter 1s backup
  server master0 master0.ocp4.example.com:22623 check inter 1s
  server master1 master1.ocp4.example.com:22623 check inter 1s
  server master2 master2.ocp4.example.com:22623 check inter 1s
listen ingress-router-443
  bind *:443
  mode tcp
  balance source
  server compute0 compute0.ocp4.example.com:443 check inter 1s
  server compute1 compute1.ocp4.example.com:443 check inter 1s
listen ingress-router-80
  bind *:80
  mode tcp
  balance source
  server compute0 compute0.ocp4.example.com:80 check inter 1s
  server compute1 compute1.ocp4.example.com:80 check inter 1s

```

where:

listen api-server-6443

Port **6443** handles the Kubernetes API traffic and points to the control plane machines.

server bootstrap bootstrap.ocp4.example.com

The bootstrap entries must be in place before the OpenShift Container Platform cluster installation and they must be removed after the bootstrap process is complete.

listen machine-config-server

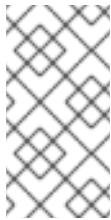
Port **22623** handles the machine config server traffic and points to the control plane machines.

listen ingress-router-443

Port **443** handles the HTTPS traffic and points to the machines that run the Ingress Controller pods. The Ingress Controller pods run on the compute machines by default.

listen ingress-router-80

Port **80** handles the HTTP traffic and points to the machines that run the Ingress Controller pods. The Ingress Controller pods run on the compute machines by default.

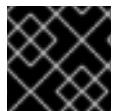


NOTE

If you are deploying a three-node cluster with zero compute nodes, the Ingress Controller pods run on the control plane nodes. In three-node cluster deployments, you must configure your application Ingress load balancer to route HTTP and HTTPS traffic to the control plane nodes.

3.2.7. Validating DNS resolution for user-provisioned infrastructure

To prevent network-related installation failures and ensure node connectivity in OpenShift Container Platform, validate your DNS configuration before deploying on user-provisioned infrastructure. This verification confirms that all required records resolve correctly, providing the stable foundation necessary for cluster communication.



IMPORTANT

The validation steps detailed in this section must succeed before you install your cluster.

Prerequisites

- You have configured the required DNS records for your user-provisioned infrastructure.

Procedure

1. From your installation node, run DNS lookups against the record names of the Kubernetes API, the wildcard routes, and the cluster nodes. Validate that the IP addresses contained in the responses correspond to the correct components.
 - a. Perform a lookup against the Kubernetes API record name. Check that the result points to the IP address of the API load balancer:

```
$ dig +noall +answer @<nameserver_ip> api.<cluster_name>.<base_domain>
```

Replace **<nameserver_ip>** with the IP address of the name server, **<cluster_name>** with your cluster name, and **<base_domain>** with your base domain name.

Example output

```
api.ocp4.example.com. 604800 IN A 192.168.1.5
```

- b. Perform a lookup against the Kubernetes internal API record name. Check that the result points to the IP address of the API load balancer:

```
$ dig +noall +answer @<nameserver_ip> api-int.<cluster_name>.<base_domain>
```

Example output

api-int.ocp4.example.com. 604800 IN A 192.168.1.5

- Test an example `*.apps.<cluster_name>.<base_domain>` DNS wildcard lookup. All of the application wildcard lookups must resolve to the IP address of the application ingress load balancer:

```
$ dig +noall +answer @<nameserver_ip> random.apps.<cluster_name>.<base_domain>
```

Example output

random.apps.ocp4.example.com. 604800 IN A 192.168.1.5



NOTE

In the example outputs, the same load balancer is used for the Kubernetes API and application ingress traffic. In production scenarios, you can deploy the API and application ingress load balancers separately so that you can scale the load balancer infrastructure for each in isolation.

You can replace **random** with another wildcard value. For example, you can query the route to the OpenShift Container Platform console:

```
$ dig +noall +answer @<nameserver_ip> console-openshift-console.apps.<cluster_name>.<base_domain>
```

Example output

console-openshift-console.apps.ocp4.example.com. 604800 IN A 192.168.1.5

- Run a lookup against the bootstrap DNS record name. Check that the result points to the IP address of the bootstrap node:

```
$ dig +noall +answer @<nameserver_ip> bootstrap.<cluster_name>.<base_domain>
```

Example output

bootstrap.ocp4.example.com. 604800 IN A 192.168.1.96

- Use this method to perform lookups against the DNS record names for the control plane and compute nodes. Check that the results correspond to the IP addresses of each node.
- From your installation node, run reverse DNS lookups against the IP addresses of the load balancer and the cluster nodes. Validate that the record names contained in the responses correspond to the correct components.
 - Perform a reverse lookup against the IP address of the API load balancer. Check that the response includes the record names for the Kubernetes API and the Kubernetes internal API:

```
$ dig +noall +answer @<nameserver_ip> -x 192.168.1.5
```

Example output

```
5.1.168.192.in-addr.arpa. 604800 IN PTR api-int.ocp4.example.com.  
5.1.168.192.in-addr.arpa. 604800 IN PTR api.ocp4.example.com.
```

where:

api-int.ocp4.example.com

Specifies the record name for the Kubernetes internal API.

api.ocp4.example.com

Specifies the record name for the Kubernetes API.

NOTE

A PTR record is not required for the OpenShift Container Platform application wildcard. No validation step is needed for reverse DNS resolution against the IP address of the application ingress load balancer.

- b. Perform a reverse lookup against the IP address of the bootstrap node. Check that the result points to the DNS record name of the bootstrap node:

```
$ dig +noall +answer @<nameserver_ip> -x 192.168.1.96
```

Example output

```
96.1.168.192.in-addr.arpa. 604800 IN PTR bootstrap.ocp4.example.com.
```

- c. Use this method to perform reverse lookups against the IP addresses for the control plane and compute nodes. Check that the results correspond to the DNS record names of each node.

3.3. INSTALLING A CLUSTER ON VSphere WITH USER-PROVISIONED INFRASTRUCTURE

In OpenShift Container Platform version 4.19, you can install a cluster on VMware vSphere infrastructure that you provision.



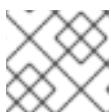
IMPORTANT

The steps for performing a user-provisioned infrastructure installation are provided as an example only. Installing a cluster with infrastructure you provide requires knowledge of the vSphere platform and the installation process of OpenShift Container Platform. Use the user-provisioned infrastructure installation instructions as a guide; you are free to create the required resources through other methods.

3.3.1. Prerequisites

- You have completed the tasks in [Preparing to install a cluster using user-provisioned infrastructure](#).

- You reviewed your VMware platform licenses. Red Hat does not place any restrictions on your VMware licenses, but some VMware infrastructure components require licensing.
- You reviewed details about the [OpenShift Container Platform installation and update processes](#).
- You read the documentation on [selecting a cluster installation method and preparing it for users](#).
- You provisioned [persistent storage](#) for your cluster. To deploy a private image registry, your storage must provide **ReadWriteMany** access modes.
- Completing the installation requires that you upload the Red Hat Enterprise Linux CoreOS (RHCOS) OVA on vSphere hosts. The machine from which you complete this process requires access to port 443 on the vCenter and ESXi hosts. You verified that port 443 is accessible.
- If you use a firewall, you confirmed with the administrator that port 443 is accessible. Control plane nodes must be able to reach vCenter and ESXi hosts on port 443 for the installation to succeed.
- If you use a firewall, you [configured it to allow the sites](#) that your cluster requires access to.



NOTE

Be sure to also review this site list if you are configuring a proxy.

3.3.2. Internet access for OpenShift Container Platform

In OpenShift Container Platform 4.19, you require access to the internet to install your cluster.

You must have internet access to perform the following actions:

- Access [OpenShift Cluster Manager](#) to download the installation program and perform subscription management. If the cluster has internet access and you do not disable Telemetry, that service automatically entitles your cluster.
- 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 required content and use it to populate a mirror registry with the installation packages. 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.

3.3.3. VMware vSphere region and zone enablement

You can deploy an OpenShift Container Platform cluster to multiple vSphere data centers. Each data center can run multiple clusters. This configuration reduces the risk of a hardware failure or network outage that can cause your cluster to fail. To enable regions and zones, you must define multiple failure domains for your OpenShift Container Platform cluster.



IMPORTANT

The VMware vSphere region and zone enablement feature requires the vSphere Container Storage Interface (CSI) driver as the default storage driver in the cluster. As a result, the feature is only available on a newly installed cluster.

For a cluster that was upgraded from a previous release, you must enable CSI automatic migration for the cluster. You can then configure multiple regions and zones for the upgraded cluster.

The default installation configuration deploys a cluster to a single vSphere data center. If you want to deploy a cluster to multiple vSphere data centers, you must create an installation configuration file that enables the region and zone feature.

The default **install-config.yaml** file includes **vcenters** and **failureDomains** fields, where you can specify multiple vSphere data centers and clusters for your OpenShift Container Platform cluster. You can leave these fields blank if you want to install an OpenShift Container Platform cluster in a vSphere environment that consists of single data center.

The following list describes terms associated with defining zones and regions for your cluster:

- Failure domain: Establishes the relationships between a region and zone. You define a failure domain by using vCenter objects, such as a **datastore** object. A failure domain defines the vCenter location for OpenShift Container Platform cluster nodes.
- Region: Specifies a vCenter data center. You define a region by using a tag from the **openshift-region** tag category.
- Zone: Specifies a vCenter cluster. You define a zone by using a tag from the **openshift-zone** tag category.



NOTE

If you plan on specifying more than one failure domain in your **install-config.yaml** file, you must create tag categories, zone tags, and region tags in advance of creating the configuration file.

You must create a vCenter tag for each vCenter data center, which represents a region. Additionally, you must create a vCenter tag for each cluster than runs in a data center, which represents a zone. After you create the tags, you must attach each tag to their respective data centers and clusters.

The following table outlines an example of the relationship among regions, zones, and tags for a configuration with multiple vSphere data centers running in a single VMware vCenter.

Data center (region)	Cluster (zone)	Tags
us-east	us-east-1	us-east-1a
		us-east-1b
	us-east-2	us-east-2a

Data center (region)	Cluster (zone)	Tags
		us-east-2b
us-west	us-west-1	us-west-1a
		us-west-1b
	us-west-2	us-west-2a
		us-west-2b

Additional resources

- [Additional VMware vSphere configuration parameters](#)
- [Deprecated VMware vSphere configuration parameters](#)
- [vSphere automatic migration](#)
- [VMware vSphere CSI Driver Operator](#)

3.3.4. Manually creating the installation configuration file

To customise your OpenShift Container Platform deployment and meet specific network requirements, manually create the installation configuration file. This ensures that the installation program uses your tailored settings rather than default values during the setup process.

Prerequisites

- You have an SSH public key on your local machine for use with the installation program. You can use the key for SSH authentication onto your cluster nodes for debugging and disaster recovery.
- You have obtained the OpenShift Container Platform installation program and the pull secret for your cluster.

Procedure

1. Create an installation directory to store your required installation assets in:

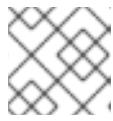
```
$ mkdir <installation_directory>
```



IMPORTANT

You must create a directory. Some installation assets, such as 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.

2. Customize the provided sample **install-config.yaml** file template and save the file in the **<installation_directory>**.



NOTE

You must name this configuration file **install-config.yaml**.

3. If you are installing a three-node cluster, modify the **install-config.yaml** file by setting the **compute.replicas** parameter to **0**. This ensures that the cluster's control planes are schedulable. For more information, see "Installing a three-node cluster on vSphere".
4. Back up the **install-config.yaml** file so that you can use it to install many clusters.



IMPORTANT

Back up the **install-config.yaml** file now, because the installation process consumes the file in the next step.

Additional resources

- [Installation configuration parameters](#)

3.3.4.1. Sample **install-config.yaml** file for VMware vSphere

You can customize the **install-config.yaml** file to specify more details about your OpenShift Container Platform cluster's platform or modify the values of the required parameters.

```
additionalTrustBundlePolicy: Proxyonly
apiVersion: v1
baseDomain: example.com 1
compute: 2
- architecture: amd64
  name: <worker_node>
  platform: {}
  replicas: 0 3
controlPlane: 4
  architecture: amd64
  name: <parent_node>
  platform: {}
  replicas: 3 5
metadata:
  creationTimestamp: null
  name: test 6
networking:
---
platform:
  vsphere:
    failureDomains: 7
    - name: <failure_domain_name>
      region: <default_region_name>
      server: <fully_qualified_domain_name>
    topology:
      computeCluster: "<data_center>/host/<cluster>"
```

```

datacenter: <data_center> 8
datastore: "/<data_center>/datastore/<datastore>" 9
networks:
- <VM_Network_name>
resourcePool: "/<data_center>/host/<cluster>/Resources/<resourcePool>" 10
folder: "/<data_center_name>/vm/<folder_name>/<subfolder_name>" 11
zone: <default_zone_name>
vcenters:
- datacenters:
- <data_center>
password: <password> 12
port: 443
server: <fullyQualifiedDomainName> 13
user: administrator@vsphere.local
diskType: thin 14
fips: false 15
pullSecret: '{"auths": ...}' 16
sshKey: 'ssh-ed25519 AAAA...' 17

```

- 1** The base domain of the cluster. All DNS records must be sub-domains of this base and include the cluster name.
- 2** **4** The **controlPlane** section is a single mapping, but the **compute** section is a sequence of mappings. To meet the requirements of the different data structures, the first line of the **compute** section must begin with a hyphen, -, and the first line of the **controlPlane** section must not. Both sections define a single machine pool, so only one control plane is used. OpenShift Container Platform does not support defining multiple compute pools.
- 3** You must set the value of the **replicas** parameter to **0**. This parameter controls the number of workers that the cluster creates and manages for you, which are functions that the cluster does not perform when you use user-provisioned infrastructure. You must manually deploy worker machines for the cluster to use before you finish installing OpenShift Container Platform.
- 5** The number of control plane machines that you add to the cluster. Because the cluster uses this values as the number of etcd endpoints in the cluster, the value must match the number of control plane machines that you deploy.
- 6** The cluster name that you specified in your DNS records.
- 7** Establishes the relationships between a region and zone. You define a failure domain by using vCenter objects, such as a **datastore** object. A failure domain defines the vCenter location for OpenShift Container Platform cluster nodes.
- 8** The vSphere data center.
- 9** The path to the vSphere datastore that holds virtual machine files, templates, and ISO images.



IMPORTANT

You can specify the path of any datastore that exists in a datastore cluster. By default, Storage vMotion is automatically enabled for a datastore cluster. Red Hat does not support Storage vMotion, so you must disable Storage vMotion to avoid data loss issues for your OpenShift Container Platform cluster.

If you must specify VMs across multiple datastores, use a **datastore** object to specify a failure domain in your cluster's **install-config.yaml** configuration file. For more information, see "VMware vSphere region and zone enablement".

- 10 Optional: For installer-provisioned infrastructure, the absolute path of an existing resource pool where the installation program creates the virtual machines, for example, `/<data_center_name>/host/<cluster_name>/Resources/<resource_pool_name>/<optional_new_resource_pool_name>`. If you do not specify a value, resources are installed in the root of the cluster `/example_data_center/host/example_cluster/Resources`.
- 11 Optional: For installer-provisioned infrastructure, the absolute path of an existing folder where the installation program creates the virtual machines, for example, `/<data_center_name>/vm/<folder_name>/<subfolder_name>`. If you do not provide this value, the installation program creates a top-level folder in the data center virtual machine folder that is named with the infrastructure ID. If you are providing the infrastructure for the cluster and you do not want to use the default **StorageClass** object, named **thin**, you can omit the **folder** parameter from the **install-config.yaml** file.
- 12 The password associated with the vSphere user.
- 13 The fully-qualified hostname or IP address of the vCenter server.



IMPORTANT

The Cloud Controller Manager Operator performs a connectivity check on a provided hostname or IP address. Ensure that you specify a hostname or an IP address to a reachable vCenter server. If you provide metadata to a non-existent vCenter server, installation of the cluster fails at the bootstrap stage.

- 14 The vSphere disk provisioning method.
- 15 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.



IMPORTANT

To enable FIPS mode for your cluster, you must run the installation program from a Red Hat Enterprise Linux (RHEL) computer configured to operate in FIPS mode. For more information about configuring FIPS mode on RHEL, see [Switching RHEL to FIPS mode](#).

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.

- 16 The pull secret that you obtained from [OpenShift Cluster Manager](#). This pull secret allows you to authenticate with the services that are provided by the included authorities, including Quay.io,
- 17 The public portion of the default SSH key for the **core** user in Red Hat Enterprise Linux CoreOS (RHCOS).

3.3.4.2. Configuring the cluster-wide proxy during installation

To enable internet access in environments that deny direct connections, configure a cluster-wide proxy in the **install-config.yaml** file. This configuration ensures that the new OpenShift Container Platform cluster routes traffic through the specified HTTP or HTTPS proxy.

Prerequisites

- You have an existing **install-config.yaml** file.
- You have reviewed the sites that your cluster requires access to and determined whether any of them need to bypass the proxy. By default, all cluster egress traffic is proxied, including calls to hosting cloud provider APIs. You added sites to the **Proxy** object's **spec.noProxy** field to bypass the proxy if necessary.



NOTE

The **Proxy** object **status.noProxy** field is populated with the values of the **networking.machineNetwork[].cidr**, **networking.clusterNetwork[].cidr**, and **networking.serviceNetwork[]** fields from your installation configuration.

For installations on Amazon Web Services (AWS), Google Cloud, Microsoft Azure, and Red Hat OpenStack Platform (RHOSP), the **Proxy** object **status.noProxy** field is also populated with the instance metadata endpoint (169.254.169.254).

Procedure

1. Edit your **install-config.yaml** file and add the proxy settings. For example:

```
apiVersion: v1
baseDomain: my.domain.com
proxy:
  httpProxy: http://<username>:<pswd>@<ip>:<port>
  httpsProxy: https://<username>:<pswd>@<ip>:<port>
  noProxy: example.com
  additionalTrustBundle: |
    -----BEGIN CERTIFICATE-----
    <MY_TRUSTED_CA_CERT>
    -----END CERTIFICATE-----
  additionalTrustBundlePolicy: <policy_to_add_additionalTrustBundle>
  # ...
```

where:

proxy.httpProxy

Specifies a proxy URL to use for creating HTTP connections outside the cluster. The URL scheme must be **http**.

proxy.httpsProxy

Specifies a proxy URL to use for creating HTTPS connections outside the cluster.

proxy.noProxy

Specifies a comma-separated list of destination domain names, IP addresses, or other network CIDRs to exclude from proxying. Preface a domain with `.` to match subdomains only.

For example, `.y.com` matches `x.y.com`, but not `y.com`. Use `*` to bypass the proxy for all destinations. You must include vCenter's IP address and the IP range that you use for its machines.

additionalTrustBundle

If provided, the installation program generates a config map that is named **user-ca-bundle** in the **openshift-config** namespace to hold the additional CA certificates. If you provide **additionalTrustBundle** and at least one proxy setting, the **Proxy** object is configured to reference the **user-ca-bundle** config map in the **trustedCA** field. The Cluster Network Operator then creates a **trusted-ca-bundle** config map that merges the contents specified for the **trustedCA** parameter with the RHCOS trust bundle. The **additionalTrustBundle** field is required unless the proxy's identity certificate is signed by an authority from the RHCOS trust bundle.

additionalTrustBundlePolicy

Specifies the policy that determines the configuration of the **Proxy** object to reference the **user-ca-bundle** config map in the **trustedCA** field. The allowed values are **Proxyonly** and **Always**. Use **Proxyonly** to reference the **user-ca-bundle** config map only when **http/https** proxy is configured. Use **Always** to always reference the **user-ca-bundle** config map. The default value is **Proxyonly**. Optional parameter.



NOTE

The installation program does not support the proxy **readinessEndpoints** field.



NOTE

If the installer times out, restart and then complete the deployment by using the **wait-for** command of the installer. For example:

+

```
$ ./openshift-install wait-for install-complete --log-level debug
```

2. Save the file and reference it when installing OpenShift Container Platform.

The installation program creates a cluster-wide proxy that is named **cluster** that uses the proxy settings in the provided **install-config.yaml** file. If no proxy settings are provided, a **cluster Proxy** object is still created, but it will have a nil **spec**.



NOTE

Only the **Proxy** object named **cluster** is supported, and no additional proxies can be created.

3.3.4.3. Configuring regions and zones for a VMware vCenter

You can modify the default installation configuration file, so that you can deploy an OpenShift Container Platform cluster to multiple vSphere data centers.

The default **install-config.yaml** file configuration from the previous release of OpenShift Container Platform is deprecated. You can continue to use the deprecated default configuration, but the **openshift-installer** will prompt you with a warning message that indicates the use of deprecated fields in the configuration file.

Prerequisites

- You have an existing **install-config.yaml** installation configuration file.



IMPORTANT

You must specify at least one failure domain for your OpenShift Container Platform cluster, so that you can provision data center objects for your VMware vCenter server. Consider specifying multiple failure domains if you need to provision virtual machine nodes in different data centers, clusters, datastores, and other components. To enable regions and zones, you must define multiple failure domains for your OpenShift Container Platform cluster.

- You have installed the **govc** command line tool.



IMPORTANT

The example uses the **govc** command. The **govc** command is an open source command available from VMware; it is not available from Red Hat. The Red Hat support team does not maintain the **govc** command. Instructions for downloading and installing **govc** are found on the VMware documentation website.

Procedure

1. Create the **openshift-region** and **openshift-zone** vCenter tag categories by running the following commands:



IMPORTANT

If you specify different names for the **openshift-region** and **openshift-zone** vCenter tag categories, the installation of the OpenShift Container Platform cluster fails.

```
$ govc tags.category.create -d "OpenShift region" openshift-region
```

```
$ govc tags.category.create -d "OpenShift zone" openshift-zone
```

2. For each region where you want to deploy your cluster, create a region tag by running the following command:

```
$ govc tags.create -c <region_tag_category> <region_tag>
```

3. For each zone where you want to deploy your cluster, create a zone tag by running the following command:

```
$ govc tags.create -c <zone_tag_category> <zone_tag>
```

4. Attach region tags to each vCenter data center object by running the following command:

```
$ govc tags.attach -c <region_tag_category> <region_tag_1> /<data_center_1>
```

5. Attach the zone tags to each vCenter cluster object by running the following command:

```
$ govc tags.attach -c <zone_tag_category> <zone_tag_1> /<data_center_1>/host/<cluster1>
```

6. Change to the directory that contains the installation program and initialize the cluster deployment according to your chosen installation requirements.

Sample `install-config.yaml` file with multiple data centers defined in a vSphere center

```
# ...
compute:
---
vsphere:
  zones:
    - "<machine_pool_zone_1>"
    - "<machine_pool_zone_2>"
# ...
controlPlane:
# ...
vsphere:
  zones:
    - "<machine_pool_zone_1>"
    - "<machine_pool_zone_2>"
# ...
platform:
  vsphere:
    vcenters:
# ...
  datacenters:
    - <data_center_1_name>
    - <data_center_2_name>
failureDomains:
  - name: <machine_pool_zone_1>
    region: <region_tag_1>
    zone: <zone_tag_1>
    server: <fullyQualifiedDomainName>
topology:
  datacenter: <data_center_1>
  computeCluster: "/<data_center_1>/host/<cluster1>"
  networks:
    - <VM_Network1_name>
    datastore: "/<data_center_1>/datastore/<datastore1>"
    resourcePool: "/<data_center_1>/host/<cluster1>/Resources/<resourcePool1>"
    folder: "/<data_center_1>/vm/<folder1>"
  - name: <machine_pool_zone_2>
    region: <region_tag_2>
```

```

zone: <zone_tag_2>
server: <fully_qualified_domain_name>
topology:
  datacenter: <data_center_2>
  computeCluster: "/<data_center_2>/host/<cluster2>" 
  networks:
    - <VM_Network2_name>
  datastore: "/<data_center_2>/datastore/<datastore2>" 
  resourcePool: "/<data_center_2>/host/<cluster2>/Resources/<resourcePool2>" 
  folder: "/<data_center_2>/vm/<folder2>" 
# ...

```

3.3.5. Creating the Kubernetes manifest and Ignition config files

To customize cluster definitions and manually start machines, generate the Kubernetes manifest and Ignition config files. These assets provide the necessary instructions to configure the cluster infrastructure according to your specific deployment requirements.

The installation configuration file transforms into the Kubernetes manifests. The manifests wrap into the Ignition configuration files, which are later used to configure the cluster machines.



IMPORTANT

- The Ignition config files that the OpenShift Container Platform 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.
- It is recommended that you use 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.

Prerequisites

- You obtained the OpenShift Container Platform installation program.
- You created the **install-config.yaml** installation configuration file.

Procedure

1. Change to the directory that contains the OpenShift Container Platform installation program and generate the Kubernetes manifests for the cluster:

```
$ ./openshift-install create manifests --dir <installation_directory>
```

where

<installation_directory>

Specifies the installation directory that contains the **install-config.yaml** file you created.

2. Remove the Kubernetes manifest files that define the control plane machines, compute machine sets, and control plane machine sets:

```
$ rm -f openshift/99_openshift-cluster-api_master-machines-*.yaml openshift/99_openshift-cluster-api_worker-machineset-*.yaml openshift/99_openshift-machine-api_master-control-plane-machine-set.yaml
```

Because you create and manage these resources yourself, you do not have to initialize them. You can preserve the compute machine set files to create compute machines by using the machine API, but you must update references to them to match your environment.



WARNING

If you are installing a three-node cluster, skip the following step to allow the control plane nodes to be schedulable.



IMPORTANT

When you configure control plane nodes from the default unschedulable to schedulable, additional subscriptions are required. This is because control plane nodes then become compute nodes.

3. Check that the **mastersSchedulable** parameter in the `<installation_directory>/manifests/cluster-scheduler-02-config.yaml` Kubernetes manifest file is set to **false**. This setting prevents pods from being scheduled on the control plane machines:
 - a. Open the `<installation_directory>/manifests/cluster-scheduler-02-config.yaml` file.
 - b. Locate the **mastersSchedulable** parameter and ensure that it is set to **false**.
 - c. Save and exit the file.
4. To create the Ignition configuration files, run the following command from the directory that contains the installation program:

```
$ ./openshift-install create ignition-configs --dir <installation_directory>
```

where:

`<installation_directory>`

Specifies the same installation directory.

Ignition config files are created for the bootstrap, control plane, and compute nodes in the installation directory. The **kubeadmin-password** and **kubeconfig** files are created in the `./<installation_directory>/auth` directory:

```

.
├── auth
│   └── kubeadmin-password

```

```

    └── kubeconfig
    ├── bootstrap.ign
    ├── master.ign
    ├── metadata.json
    └── worker.ign

```

3.3.6. Extracting the infrastructure name

The Ignition config files contain a unique cluster identifier that you can use to uniquely identify your cluster in VMware vSphere. If you plan to use the cluster identifier as the name of your virtual machine folder, you must extract it.

Prerequisites

- You obtained the OpenShift Container Platform installation program and the pull secret for your cluster.
- You generated the Ignition config files for your cluster.
- You installed the **jq** package.

Procedure

- To extract and view the infrastructure name from the Ignition config file metadata, run the following command:

```
$ jq -r .infralD <installation_directory>/metadata.json ①
```

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

Example output

```
openshift-vw9j6 ①
```

① The output of this command is your cluster name and a random string.

3.3.7. Installing RHCOS and starting the OpenShift Container Platform bootstrap process

To install OpenShift Container Platform on user-provisioned infrastructure on VMware vSphere, you must install Red Hat Enterprise Linux CoreOS (RHCOS) on vSphere hosts. When you install RHCOS, you must provide the Ignition config file that was generated by the OpenShift Container Platform installation program for the type of machine you are installing. If you have configured suitable networking, DNS, and load balancing infrastructure, the OpenShift Container Platform bootstrap process begins automatically after the RHCOS machines have rebooted.

Prerequisites

- You have obtained the Ignition config files for your cluster.

- You have access to an HTTP server that you can access from your computer and that the machines that you create can access.
- You have created a [vSphere cluster](#).

Procedure

1. Upload the bootstrap Ignition config file, which is named **<installation_directory>/bootstrap.ign**, that the installation program created to your HTTP server. Note the URL of this file.
2. Save the following secondary Ignition config file for your bootstrap node to your computer as **<installation_directory>/merge-bootstrap.ign**:

```
{
  "ignition": {
    "config": {
      "merge": [
        {
          "source": "<bootstrap_ignition_config_url>", ①
          "verification": {}
        }
      ],
      "timeouts": {},
      "version": "3.2.0"
    },
    "networkd": {},
    "passwd": {},
    "storage": {},
    "systemd": {}
  }
}
```

- 1 Specify the URL of the bootstrap Ignition config file that you hosted.

When you create the virtual machine (VM) for the bootstrap machine, you use this Ignition config file.

3. Locate the following Ignition config files that the installation program created:
 - **<installation_directory>/master.ign**
 - **<installation_directory>/worker.ign**
 - **<installation_directory>/merge-bootstrap.ign**
4. Convert the Ignition config files to Base64 encoding. Later in this procedure, you must add these files to the extra configuration parameter **guestinfo.ignition.config.data** in your VM. For example, if you use a Linux operating system, you can use the **base64** command to encode the files.

```
$ base64 -w0 <installation_directory>/master.ign > <installation_directory>/master.64
```

```
$ base64 -w0 <installation_directory>/worker.ign > <installation_directory>/worker.64
```

```
$ base64 -w0 <installation_directory>/merge-bootstrap.ign > <installation_directory>/merge-bootstrap.64
```



IMPORTANT

If you plan to add more compute machines to your cluster after you finish installation, do not delete these files.

5. Obtain the RHCOS OVA image. Images are available from the [RHCOS image mirror](#) page.



IMPORTANT

The RHCOS images might not change with every release of OpenShift Container Platform. You must download an image with the highest version that is less than or equal to the OpenShift Container Platform version that you install. Use the image version that matches your OpenShift Container Platform version if it is available.

The filename contains the OpenShift Container Platform version number in the format **rhcoss-vmware.<architecture>.ova**.

6. In the vSphere Client, create a folder in your data center to store your VMs.
 - a. Click the **VMs and Templates** view.
 - b. Right-click the name of your data center.
 - c. Click **New Folder** → **New VM and Template Folder**.
 - d. In the window that is displayed, enter the folder name. If you did not specify an existing folder in the **install-config.yaml** file, then create a folder with the same name as the infrastructure ID. You use this folder name so vCenter dynamically provisions storage in the appropriate location for its Workspace configuration.
7. In the vSphere Client, create a template for the OVA image and then clone the template as needed.



NOTE

In the following steps, you create a template and then clone the template for all of your cluster machines. You then provide the location for the Ignition config file for that cloned machine type when you provision the VMs.

- a. From the **Hosts and Clusters** tab, right-click your cluster name and select **Deploy OVF Template**.
- b. On the **Select an OVF** tab, specify the name of the RHCOS OVA file that you downloaded.
- c. On the **Select a name and folder** tab, set a **Virtual machine name** for your template, such as **Template-RHCOS**. Click the name of your vSphere cluster and select the folder you created in the previous step.
- d. On the **Select a compute resource** tab, click the name of your vSphere cluster.

- e. On the **Select storage** tab, configure the storage options for your VM.
 - Select **Thin Provision** or **Thick Provision**, based on your storage preferences.
 - Select the datastore that you specified in your **install-config.yaml** file.
 - If you want to encrypt your virtual machines, select **Encrypt this virtual machine**. See the section titled "Requirements for encrypting virtual machines" for more information.
- f. On the **Select network** tab, specify the network that you configured for the cluster, if available.
- g. When creating the OVF template, do not specify values on the **Customize template** tab or configure the template any further.



IMPORTANT

Do not start the original VM template. The VM template must remain off and must be cloned for new RHCOS machines. Starting the VM template configures the VM template as a VM on the platform, which prevents it from being used as a template that compute machine sets can apply configurations to.

8. Optional: Update the configured virtual hardware version in the VM template, if necessary. Follow [Upgrading a virtual machine to the latest hardware version](#) in the VMware documentation for more information.



IMPORTANT

It is recommended that you update the hardware version of the VM template to version 15 before creating VMs from it, if necessary. Using hardware version 13 for your cluster nodes running on vSphere is now deprecated. If your imported template defaults to hardware version 13, you must ensure that your ESXi host is on 6.7U3 or later before upgrading the VM template to hardware version 15. If your vSphere version is less than 6.7U3, you can skip this upgrade step; however, a future version of OpenShift Container Platform is scheduled to remove support for hardware version 13 and vSphere versions less than 6.7U3.

9. After the template deploys, deploy a VM for a machine in the cluster.
 - a. Right-click the template name and click **Clone** → **Clone to Virtual Machine**
 - b. On the **Select a name and folder** tab, specify a name for the VM. You might include the machine type in the name, such as **control-plane-0** or **compute-1**.



NOTE

Ensure that all virtual machine names across a vSphere installation are unique.

- c. On the **Select a name and folder** tab, select the name of the folder that you created for the cluster.
- d. On the **Select a compute resource** tab, select the name of a host in your data center.

- e. On the **Select clone options** tab, select **Customize this virtual machine's hardware**
- f. On the **Customize hardware** tab, click **Advanced Parameters**.



IMPORTANT

The following configuration suggestions are for example purposes only. As a cluster administrator, you must configure resources according to the resource demands placed on your cluster. To best manage cluster resources, consider creating a resource pool from the cluster's root resource pool.

- Optional: Override default DHCP networking in vSphere. To enable static IP networking:
 - Set your static IP configuration:

Example command

```
$ export IPCFG="ip=<ip>::<gateway>:<netmask>:<hostname>:<iface>:none
nameserver=srv1 [nameserver=srv2 [nameserver=srv3 [...]]]"
```

Example command

```
$ export IPCFG="ip=192.168.100.101::192.168.100.254:255.255.255.0::none
nameserver=8.8.8.8"
```

- Set the **guestinfo.afterburn.initrd.network-kargs** property before you boot a VM from an OVA in vSphere:

Example command

```
$ govc vm.change -vm "<vm_name>" -e "guestinfo.afterburn.initrd.network-
kargs=${IPCFG}"
```

- Add the following configuration parameter names and values by specifying data in the **Attribute** and **Values** fields. Ensure that you select the **Add** button for each parameter that you create.
 - **guestinfo.ignition.config.data**: Locate the base-64 encoded files that you created previously in this procedure, and paste the contents of the base64-encoded Ignition config file for this machine type.
 - **guestinfo.ignition.config.data.encoding**: Specify **base64**.
 - **disk.EnableUUID**: Specify **TRUE**.
 - **stealclock.enable**: If this parameter was not defined, add it and specify **TRUE**.
 - Create a child resource pool from the cluster's root resource pool. Perform resource allocation in this child resource pool.

- g. In the **Virtual Hardware** panel of the **Customize hardware** tab, modify the specified values as required. Ensure that the amount of RAM, CPU, and disk storage meets the minimum requirements for the machine type.

- h. Complete the remaining configuration steps. On clicking the **Finish** button, you have completed the cloning operation.
- i. From the **Virtual Machines** tab, right-click on your VM and then select **Power → Power On**.
- j. Check the console output to verify that Ignition ran.

Example command

```
Ignition: ran on 2022/03/14 14:48:33 UTC (this boot)
Ignition: user-provided config was applied
```

Next steps

- Create the rest of the machines for your cluster by following the preceding steps for each machine.



IMPORTANT

You must create the bootstrap and control plane machines at this time. Because some pods are deployed on compute machines by default, also create at least two compute machines before you install the cluster.

3.3.8. Adding more compute machines to a cluster in vSphere

You can add more compute machines to a user-provisioned OpenShift Container Platform cluster on VMware vSphere.

After your vSphere template deploys in your OpenShift Container Platform cluster, you can deploy a virtual machine (VM) for a machine in that cluster.



NOTE

If you are installing a three-node cluster, skip this step. A three-node cluster consists of three control plane machines, which also act as compute machines.

Prerequisites

- Obtain the base64-encoded Ignition file for your compute machines.
- You have access to the vSphere template that you created for your cluster.

Procedure

1. Right-click the template's name and click **Clone → Clone to Virtual Machine**
2. On the **Select a name and folder** tab, specify a name for the VM. You might include the machine type in the name, such as **compute-1**.



NOTE

Ensure that all virtual machine names across a vSphere installation are unique.

3. On the **Select a name and folder** tab, select the name of the folder that you created for the cluster.
4. On the **Select a compute resource** tab, select the name of a host in your data center.
5. On the **Select storage** tab, select storage for your configuration and disk files.
6. On the **Select clone options** tab, select **Customize this virtual machine's hardware**.
7. On the **Customize hardware** tab, click **Advanced Parameters**.
 - Add the following configuration parameter names and values by specifying data in the **Attribute** and **Values** fields. Ensure that you select the **Add** button for each parameter that you create.
 - **guestinfo.ignition.config.data**: Paste the contents of the base64-encoded compute Ignition config file for this machine type.
 - **guestinfo.ignition.config.data.encoding**: Specify **base64**.
 - **disk.EnableUUID**: Specify **TRUE**.
8. In the **Virtual Hardware** panel of the **Customize hardware** tab, modify the specified values as required. Ensure that the amount of RAM, CPU, and disk storage meets the minimum requirements for the machine type. If many networks exist, select **Add New Device** > **Network Adapter**, and then enter your network information in the fields provided by the **New Network** menu item.
9. Complete the remaining configuration steps. On clicking the **Finish** button, you have completed the cloning operation.
10. From the **Virtual Machines** tab, right-click on your VM and then select **Power** → **Power On**.

Next steps

- Continue to create more compute machines for your cluster.

3.3.9. Disk partitioning

In most cases, data partitions are originally created by installing RHCOS, rather than by installing another operating system. In such cases, the OpenShift Container Platform installer should be allowed to configure your disk partitions.

However, there are two cases where you might want to intervene to override the default partitioning when installing an OpenShift Container Platform node:

- Create separate partitions: For greenfield installations on an empty disk, you might want to add separate storage to a partition. This is officially supported for making **/var** or a subdirectory of **/var**, such as **/var/lib/etcd**, a separate partition, but not both.



IMPORTANT

For disk sizes larger than 100GB, and especially disk sizes larger than 1TB, create a separate **/var** partition. See "Creating a separate **/var** partition" and this [Red Hat Knowledgebase article](#) for more information.



IMPORTANT

Kubernetes supports only two file system partitions. If you add more than one partition to the original configuration, Kubernetes cannot monitor all of them.

- **Retain existing partitions:** For a brownfield installation where you are reinstalling OpenShift Container Platform on an existing node and want to retain data partitions installed from your previous operating system, there are both boot arguments and options to **coreos-installer** that allow you to retain existing data partitions.

3.3.10. Creating a separate `/var` partition

In general, disk partitioning for OpenShift Container Platform should be left to the installer. However, there are cases where you might want to create separate partitions in a part of the filesystem that you expect to grow.

OpenShift Container Platform supports the addition of a single partition to attach storage to either the `/var` partition or a subdirectory of `/var`. For example:

- **/var/lib/containers:** Holds container-related content that can grow as more images and containers are added to a system.
- **/var/lib/etcd:** Holds data that you might want to keep separate for purposes such as performance optimization of etcd storage.
- **/var:** Holds data that you might want to keep separate for purposes such as auditing.



IMPORTANT

For disk sizes larger than 100GB, and especially larger than 1TB, create a separate `/var` partition.

Storing the contents of a `/var` directory separately makes it easier to grow storage for those areas as needed and reinstall OpenShift Container Platform at a later date and keep that data intact. With this method, you will not have to pull all your containers again, nor will you have to copy massive log files when you update systems.

Because `/var` must be in place before a fresh installation of Red Hat Enterprise Linux CoreOS (RHCOS), the following procedure sets up the separate `/var` partition by creating a machine config manifest that is inserted during the **openshift-install** preparation phases of an OpenShift Container Platform installation.

Procedure

1. Create a directory to hold the OpenShift Container Platform installation files:

```
$ mkdir $HOME/clusterconfig
```

2. Run **openshift-install** to create a set of files in the `manifest` and `openshift` subdirectories. Answer the system questions as you are prompted:

```
$ openshift-install create manifests --dir $HOME/clusterconfig
? SSH Public Key ...
```

```
$ ls $HOME/clusterconfig/openshift/
99_kubeadmin-password-secret.yaml
99_openshift-cluster-api_master-machines-0.yaml
99_openshift-cluster-api_master-machines-1.yaml
99_openshift-cluster-api_master-machines-2.yaml
...
```

3. Create a Butane config that configures the additional partition. For example, name the file **\$HOME/clusterconfig/98-var-partition.bu**, change the disk device name to the name of the storage device on the **worker** systems, and set the storage size as appropriate. This example places the **/var** directory on a separate partition:

```
variant: openshift
version: 4.19.0
metadata:
  labels:
    machineconfiguration.openshift.io/role: worker
  name: 98-var-partition
storage:
  disks:
    - device: /dev/disk/by-id/<device_name> ①
      partitions:
        - label: var
          start_mib: <partition_start_offset> ②
          size_mib: <partition_size> ③
          number: 5
  filesystems:
    - device: /dev/disk/by-partlabel/var
      path: /var
      format: xfs
      mount_options: [defaults, prjquota] ④
      with_mount_unit: true
```

- 1 The storage device name of the disk that you want to partition.
- 2 When adding a data partition to the boot disk, a minimum value of 25000 mebibytes is recommended. The root file system is automatically resized to fill all available space up to the specified offset. If no value is specified, or if the specified value is smaller than the recommended minimum, the resulting root file system will be too small, and future reinstalls of RHCOS might overwrite the beginning of the data partition.
- 3 The size of the data partition in mebibytes.
- 4 The **prjquota** mount option must be enabled for filesystems used for container storage.



NOTE

When creating a separate **/var** partition, you cannot use different instance types for worker nodes, if the different instance types do not have the same device name.

4. Create a manifest from the Butane config and save it to the **clusterconfig/openshift** directory. For example, run the following command:

```
$ butane $HOME/clusterconfig/98-var-partition.bu -o $HOME/clusterconfig/openshift/98-var-partition.yaml
```

5. Run **openshift-install** again to create Ignition configs from a set of files in the **manifest** and **openshift** subdirectories:

```
$ openshift-install create ignition-configs --dir $HOME/clusterconfig
$ ls $HOME/clusterconfig/
auth bootstrap.ign master.ign metadata.json worker.ign
```

Now you can use the Ignition config files as input to the vSphere installation procedures to install Red Hat Enterprise Linux CoreOS (RHCOS) systems.

3.3.11. Waiting for the bootstrap process to complete

To install OpenShift Container Platform, use Ignition configuration files to initialize the bootstrap process after the cluster nodes boot into RHCOS. You must wait for this process to complete to ensure the cluster is fully installed.

Prerequisites

- You have created the Ignition config files for your cluster.
- You have configured suitable network, DNS, and load balancing infrastructure.
- You have obtained the installation program and generated the Ignition config files for your cluster.
- You installed RHCOS on your cluster machines and provided the Ignition config files that the OpenShift Container Platform installation program generated.
- Your machines have direct internet access or have an HTTP or HTTPS proxy available.

Procedure

1. Monitor the bootstrap process:

```
$ ./openshift-install --dir <installation_directory> wait-for bootstrap-complete \
--log-level=info
```

where:

<installation_directory>

Specifies the path to the directory that stores the installation files.

--log-level=info

Specifies **warn**, **debug**, or **error** instead of **info** to view different installation details.

Example output

```
INFO Waiting up to 30m0s for the Kubernetes API at https://api.test.example.com:6443...
INFO API v1.32.3 up
INFO Waiting up to 30m0s for bootstrapping to complete...
INFO It is now safe to remove the bootstrap resources
```

The command succeeds when the Kubernetes API server signals that it has been bootstrapped on the control plane machines.

2. After the bootstrap process is complete, remove the bootstrap machine from the load balancer.



IMPORTANT

You must remove the bootstrap machine from the load balancer at this point. You can also remove or reformat the bootstrap machine itself.

3.3.12. Logging in to the cluster by using the CLI

To log in to your cluster as the default system user, export the **kubeconfig** file. This configuration enables the CLI to authenticate and connect to the specific API server created during OpenShift Container Platform installation.

The **kubeconfig** file is specific to a cluster and is created during OpenShift Container Platform installation.

Prerequisites

- You deployed an OpenShift Container Platform cluster.
- You installed the OpenShift CLI (**oc**).

Procedure

1. Export the **kubeadmin** credentials by running the following command:

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

where:

<installation_directory>

Specifies the path to the directory that stores the installation files.

2. Verify you can run **oc** commands successfully using the exported configuration by running the following command:

```
$ oc whoami
```

Example output

```
system:admin
```

3.3.13. Approving the certificate signing requests for your machines

To add machines to a cluster, verify the status of the certificate signing requests (CSRs) generated for each machine. If manual approval is required, approve the client requests first, followed by the server requests.

Prerequisites

- You added machines to your cluster.

Procedure

1. Confirm that the cluster recognizes the machines:

```
$ oc get nodes
```

Example output

NAME	STATUS	ROLES	AGE	VERSION
master-0	Ready	master	63m	v1.32.3
master-1	Ready	master	63m	v1.32.3
master-2	Ready	master	64m	v1.32.3

The output lists all of the machines that you created.



NOTE

The preceding output might not include the compute nodes, also known as worker nodes, until some CSRs are approved.

2. Review the pending CSRs and ensure that you see the client requests with the **Pending** or **Approved** status for each machine that you added to the cluster:

```
$ oc get csr
```

Example output

NAME	AGE	REQUESTOR	CONDITION
csr-8b2br	15m	system:serviceaccount:openshift-machine-config-operator:node-bootstrapper	Pending
csr-8vnps	15m	system:serviceaccount:openshift-machine-config-operator:node-bootstrapper	Pending
...			

In this example, two machines are joining the cluster. You might see more approved CSRs in the list.

3. If the CSRs were not approved, after all of the pending CSRs for the machines you added are in **Pending** status, approve the CSRs for your cluster machines:



NOTE

Because the CSRs rotate automatically, approve your CSRs within an hour of adding the machines to the cluster. If you do not approve them within an hour, the certificates will rotate, and more than two certificates will be present for each node. You must approve all of these certificates. After the client CSR is approved, the Kubelet creates a secondary CSR for the serving certificate, which requires manual approval. Then, subsequent serving certificate renewal requests are automatically approved by the **machine-approvers** if the Kubelet requests a new certificate with identical parameters.



NOTE

For clusters running on platforms that are not machine API enabled, such as bare metal and other user-provisioned infrastructure, you must implement a method of automatically approving the kubelet serving certificate requests (CSRs). If a request is not approved, then the **oc exec**, **oc rsh**, and **oc logs** commands cannot succeed, because a serving certificate is required when the API server connects to the kubelet. Any operation that contacts the Kubelet endpoint requires this certificate approval to be in place. The method must watch for new CSRs, confirm that the CSR was submitted by the **node-bootstrapper** service account in the **system:node** or **system:admin** groups, and confirm the identity of the node.

- To approve them individually, run the following command for each valid CSR:

```
$ oc adm certificate approve <csr_name>
```

where:

<csr_name>

Specifies the name of a CSR from the list of current CSRs.

- To approve all pending CSRs, run the following command:

```
$ oc get csr -o go-template='{{range .items}}{{if not .status}}{{.metadata.name}}\n{{end}}{{end}}' | xargs --no-run-if-empty oc adm certificate approve
```



NOTE

Some Operators might not become available until some CSRs are approved.

4. Now that your client requests are approved, you must review the server requests for each machine that you added to the cluster:

```
$ oc get csr
```

Example output

NAME	AGE	REQUESTOR	CONDITION
csr-bfd72	5m26s	system:node:ip-10-0-50-126.us-east-2.compute.internal	Pending

```
csr-c57lv 5m26s system:node:ip-10-0-95-157.us-east-2.compute.internal
Pending
...
```

5. If the remaining CSRs are not approved, and are in the **Pending** status, approve the CSRs for your cluster machines:

- To approve them individually, run the following command for each valid CSR:

```
$ oc adm certificate approve <csr_name>
```

where:

<csr_name>

Specifies the name of a CSR from the list of current CSRs.

- To approve all pending CSRs, run the following command:

```
$ oc get csr -o go-template='{{range .items}}{{if not .status}}{{.metadata.name}}\n{{end}}{{end}}' | xargs oc adm certificate approve
```

6. After all client and server CSRs have been approved, the machines have the **Ready** status. Verify this by running the following command:

```
$ oc get nodes
```

Example output

NAME	STATUS	ROLES	AGE	VERSION
master-0	Ready	master	73m	v1.32.3
master-1	Ready	master	73m	v1.32.3
master-2	Ready	master	74m	v1.32.3
worker-0	Ready	worker	11m	v1.32.3
worker-1	Ready	worker	11m	v1.32.3



NOTE

It can take a few minutes after approval of the server CSRs for the machines to transition to the **Ready** status.

3.3.14. Initial Operator configuration

To ensure all Operators become available, configure the required Operators immediately after the control plane initialises. This configuration is essential for stabilizing the cluster environment following the installation.

Prerequisites

- Your control plane has initialized.

Procedure

1. Watch the cluster components come online:

```
$ watch -n5 oc get clusteroperators
```

Example output

NAME SINCE	VERSION	AVAILABLE	PROGRESSING	DEGRADED
authentication	4.19.0	True	False	False 19m
baremetal	4.19.0	True	False	False 37m
cloud-credential	4.19.0	True	False	False 40m
cluster-autoscaler	4.19.0	True	False	False 37m
config-operator	4.19.0	True	False	False 38m
console	4.19.0	True	False	False 26m
csi-snapshot-controller	4.19.0	True	False	False 37m
dns	4.19.0	True	False	False 37m
etcd	4.19.0	True	False	False 36m
image-registry	4.19.0	True	False	False 31m
ingress	4.19.0	True	False	False 30m
insights	4.19.0	True	False	False 31m
kube-apiserver	4.19.0	True	False	False 26m
kube-controller-manager	4.19.0	True	False	False 36m
kube-scheduler	4.19.0	True	False	False 36m
kube-storage-version-migrator	4.19.0	True	False	False 37m
machine-api	4.19.0	True	False	False 29m
machine-approver	4.19.0	True	False	False 37m
machine-config	4.19.0	True	False	False 36m
marketplace	4.19.0	True	False	False 37m
monitoring	4.19.0	True	False	False 29m
network	4.19.0	True	False	False 38m
node-tuning	4.19.0	True	False	False 37m
openshift-apiserver	4.19.0	True	False	False 32m
openshift-controller-manager	4.19.0	True	False	False 30m
openshift-samples	4.19.0	True	False	False 32m
operator-lifecycle-manager	4.19.0	True	False	False 37m
operator-lifecycle-manager-catalog	4.19.0	True	False	False 37m
operator-lifecycle-manager-packageserver	4.19.0	True	False	False 32m
service-ca	4.19.0	True	False	False 38m
storage	4.19.0	True	False	False 37m

- Configure the Operators that are not available.

3.3.14.1. Image registry removed during installation

On platforms that do not provide shareable object storage, the OpenShift Image Registry Operator bootstraps itself as **Removed**. This allows **openshift-installer** to complete installations on these platform types.

After installation, you must edit the Image Registry Operator configuration to switch the **managementState** from **Removed** to **Managed**. When this has completed, you must configure storage.

3.3.14.2. Image registry storage configuration

The Image Registry Operator is not initially available for platforms that do not provide default storage. After installation, you must configure your registry to use storage so that the Registry Operator is made available.

Configure a persistent volume, which is required for production clusters. Where applicable, you can configure an empty directory as the storage location for non-production clusters.

You can also allow the image registry to use block storage types by using the **Recreate** rollout strategy during upgrades.

3.3.14.2.1. Configuring registry storage for VMware vSphere

As a cluster administrator, following installation you must configure your registry to use storage.

Prerequisites

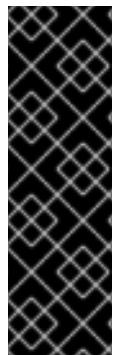
- Cluster administrator permissions.
- A cluster on VMware vSphere.
- Persistent storage provisioned for your cluster, such as Red Hat OpenShift Data Foundation.



IMPORTANT

OpenShift Container Platform supports **ReadWriteOnce** access for image registry storage when you have only one replica. **ReadWriteOnce** access also requires that the registry uses the **Recreate** rollout strategy. To deploy an image registry that supports high availability with two or more replicas, **ReadWriteMany** access is required.

- Must have "100Gi" capacity.



IMPORTANT

Testing shows issues with using the NFS server on RHEL as storage backend for core services. This includes the OpenShift Container Registry and Quay, Prometheus for monitoring storage, and Elasticsearch for logging storage. Therefore, using RHEL NFS to back PVs used by core services is not recommended.

Other NFS implementations on the marketplace might not have these issues. Contact the individual NFS implementation vendor for more information on any testing that was possibly completed against these OpenShift Container Platform core components.

Procedure

1. Change the **spec.storage.pvc** field in the **configs.imageregistry/cluster** resource.



NOTE

When you use shared storage, review your security settings to prevent outside access.

2. Verify that you do not have a registry pod by running the following command:

```
$ oc get pod -n openshift-image-registry -l docker-registry=default
```

Example output

No resources found in openshift-image-registry namespace



NOTE

If you do have a registry pod in your output, you do not need to continue with this procedure.

3. Check the registry configuration by running the following command:

```
$ oc edit configs.imageregistry.operator.openshift.io
```

Example output

```
storage:
pvc:
claim:
```

Leave the **claim** field blank to allow the automatic creation of an **image-registry-storage** persistent volume claim (PVC). The PVC is generated based on the default storage class. However, be aware that the default storage class might provide ReadWriteOnce (RWO) volumes, such as a RADOS Block Device (RBD), which can cause issues when you replicate to more than one replica.

4. Check the **clusteroperator** status by running the following command:

```
$ oc get clusteroperator image-registry
```

Example output

NAME	VERSION	AVAILABLE	PROGRESSING	DEGRADED	SINCE
MESSAGE					
image-registry	4.7	True	False	False	6h50m

3.3.14.2.2. Configuring storage for the image registry in non-production clusters

You must configure storage for the Image Registry Operator. For non-production clusters, you can set the image registry to an empty directory. If you do so, all images are lost if you restart the registry.

Procedure

- To set the image registry storage to an empty directory:

```
$ oc patch configs.imageregistry.operator.openshift.io cluster --type merge --patch '{"spec": {"storage":{"emptyDir":{}}}}'
```



WARNING

Configure this option only for non-production clusters.

If you run this command before the Image Registry Operator initializes its components, the **oc patch** command fails with the following error:

Error from server (NotFound): configs.imageregistry.operator.openshift.io "cluster" not found

Wait a few minutes and run the command again.

3.3.14.2.3. Configuring block registry storage for VMware vSphere

To allow the image registry to use block storage types such as vSphere Virtual Machine Disk (VMDK) during upgrades as a cluster administrator, you can use the **Recreate** rollout strategy.



IMPORTANT

Block storage volumes are supported but not recommended for use with image registry on production clusters. An installation where the registry is configured on block storage is not highly available because the registry cannot have more than one replica.

Procedure

1. Enter the following command to set the image registry storage as a block storage type, patch the registry so that it uses the **Recreate** rollout strategy, and runs with only **1** replica:

```
$ oc patch config.imageregistry.operator.openshift.io/cluster --type=merge -p '{"spec": {"rolloutStrategy":"Recreate", "replicas":1}}'
```

2. Provision the persistent volume (PV) for the block storage device, and create a persistent volume claim (PVC) for that volume. The requested block volume uses the ReadWriteOnce (RWO) access mode.
 - a. Create a **pvc.yaml** file with the following contents to define a VMware vSphere **PersistentVolumeClaim** object:

```
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: image-registry-storage
  namespace: openshift-image-registry
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 100Gi
```

where:

name

Specifies a unique name that represents the **PersistentVolumeClaim** object.

namespace

Specifies the **namespace** for the **PersistentVolumeClaim** object, which is **openshift-image-registry**.

accessModes

Specifies the access mode of the persistent volume claim. With **ReadWriteOnce**, the volume can be mounted with read and write permissions by a single node.

storage

The size of the persistent volume claim.

- b. Enter the following command to create the **PersistentVolumeClaim** object from the file:

```
$ oc create -f pvc.yaml -n openshift-image-registry
```

3. Enter the following command to edit the registry configuration so that it references the correct PVC:

```
$ oc edit config.imageregistry.operator.openshift.io -o yaml
```

Example output

```
storage:
pvc:
claim:
```

By creating a custom PVC, you can leave the **claim** field blank for the default automatic creation of an **image-registry-storage** PVC.

For instructions about configuring registry storage so that it references the correct PVC, see [Configuring the registry for vSphere](#).

3.3.15. Completing installation on user-provisioned infrastructure

To finalize the installation on user-provisioned infrastructure, complete the cluster deployment after configuring the Operators. This ensures the cluster is fully operational on the infrastructure that you provide.

Prerequisites

- Your control plane has initialized.
- You have completed the initial Operator configuration.

Procedure

1. Confirm that all the cluster components are online with the following command:

```
$ watch -n5 oc get clusteroperators
```

Example output

NAME SINCE	VERSION	AVAILABLE	PROGRESSING	DEGRADED
authentication	4.19.0	True	False	False 19m
baremetal	4.19.0	True	False	False 37m
cloud-credential	4.19.0	True	False	False 40m
cluster-autoscaler	4.19.0	True	False	False 37m
config-operator	4.19.0	True	False	False 38m
console	4.19.0	True	False	False 26m
csi-snapshot-controller	4.19.0	True	False	False 37m
dns	4.19.0	True	False	False 37m
etcd	4.19.0	True	False	False 36m
image-registry	4.19.0	True	False	False 31m
ingress	4.19.0	True	False	False 30m
insights	4.19.0	True	False	False 31m
kube-apiserver	4.19.0	True	False	False 26m
kube-controller-manager	4.19.0	True	False	False 36m
kube-scheduler	4.19.0	True	False	False 36m
kube-storage-version-migrator	4.19.0	True	False	False 37m
machine-api	4.19.0	True	False	False 29m
machine approver	4.19.0	True	False	False 37m
machine-config	4.19.0	True	False	False 36m
marketplace	4.19.0	True	False	False 37m
monitoring	4.19.0	True	False	False 29m
network	4.19.0	True	False	False 38m
node-tuning	4.19.0	True	False	False 37m
openshift-apiserver	4.19.0	True	False	False 32m
openshift-controller-manager	4.19.0	True	False	False 30m
openshift-samples	4.19.0	True	False	False 32m
operator-lifecycle-manager	4.19.0	True	False	False 37m
operator-lifecycle-manager-catalog	4.19.0	True	False	False 37m
operator-lifecycle-manager-packageserver	4.19.0	True	False	False 32m
service-ca	4.19.0	True	False	False 38m
storage	4.19.0	True	False	False 37m

Alternatively, the following command notifies you when all of the clusters are available. The command also retrieves and displays credentials:

```
$ ./openshift-install --dir <installation_directory> wait-for install-complete
```

where:

<installation_directory>

Specifies the path to the directory that you stored the installation files in.

Example output

```
INFO Waiting up to 30m0s for the cluster to initialize...
```

The command succeeds when the Cluster Version Operator finishes deploying the OpenShift Container Platform cluster from Kubernetes API server.



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.
- It is recommended that you use 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.

2. Confirm that the Kubernetes API server is communicating with the pods.

a. To view a list of all pods, use the following command:

```
$ oc get pods --all-namespaces
```

Example output

NAMESPACE	NAME	READY	STATUS
RESTARTS	AGE		
openshift-apiserver-operator	openshift-apiserver-operator-85cb746d55-zqhs8	1/1	
Running 1 9m			
openshift-apiserver	apiserver-67b9g	1/1	Running 0
3m			
openshift-apiserver	apiserver-ljcmx	1/1	Running 0
1m			
openshift-apiserver	apiserver-z25h4	1/1	Running 0
2m			
openshift-authentication-operator	authentication-operator-69d5d8bf84-vh2n8	1/1	
Running 0 5m			

b. View the logs for a pod that is listed in the output of the previous command by using the following command:

```
$ oc logs <pod_name> -n <namespace>
```

where:

<namespace>

Specifies the pod name and namespace, as shown in the output of an earlier command. If the pod logs display, the Kubernetes API server can communicate with the cluster machines.

3. For an installation with Fibre Channel Protocol (FCP), additional steps are required to enable multipathing. Do not enable multipathing during installation.

See "Enabling multipathing with kernel arguments on RHCOS" in the *Postinstallation machine configuration tasks* documentation for more information.

You can add extra compute machines after the cluster installation is completed by following [Adding compute machines to vSphere](#).

3.3.16. Configuring vSphere DRS anti-affinity rules for control plane nodes

vSphere Distributed Resource Scheduler (DRS) anti-affinity rules can be configured to support higher availability of OpenShift Container Platform Control Plane nodes. Anti-affinity rules ensure that the vSphere Virtual Machines for the OpenShift Container Platform Control Plane nodes are not scheduled to the same vSphere Host.



IMPORTANT

- The following information applies to compute DRS only and does not apply to storage DRS.
- The **govc** command is an open-source command available from VMware; it is not available from Red Hat. The **govc** command is not supported by the Red Hat support.
- Instructions for downloading and installing **govc** are found on the VMware documentation website.

Create an anti-affinity rule by running the following command:

Example command

```
$ govc cluster.rule.create \
  -name openshift4-control-plane-group \
  -dc MyDatacenter -cluster MyCluster \
  -enable \
  -anti-affinity master-0 master-1 master-2
```

After creating the rule, your control plane nodes are automatically migrated by vSphere so they are not running on the same hosts. This might take some time while vSphere reconciles the new rule. Successful command completion is shown in the following procedure.



NOTE

The migration occurs automatically and might cause brief OpenShift API outage or latency until the migration finishes.

The vSphere DRS anti-affinity rules need to be updated manually in the event of a control plane VM name change or migration to a new vSphere Cluster.

Procedure

1. Remove any existing DRS anti-affinity rule by running the following command:

```
$ govc cluster.rule.remove \
```

```
-name openshift4-control-plane-group \
-dc MyDatacenter -cluster MyCluster
```

Example Output

```
[13-10-22 09:33:24] Reconfigure /MyDatacenter/host/MyCluster...OK
```

2. Create the rule again with updated names by running the following command:

```
$ govc cluster.rule.create \
-name openshift4-control-plane-group \
-dc MyDatacenter -cluster MyOtherCluster \
-enable \
-anti-affinity master-0 master-1 master-2
```

3.3.17. Telemetry access for OpenShift Container Platform

To provide metrics about cluster health and the success of updates, the Telemetry service requires internet access. When connected, this service runs automatically by default and registers your cluster to [OpenShift Cluster Manager](#).

After you confirm that your [OpenShift Cluster Manager](#) inventory is correct, either maintained automatically by Telemetry or manually by using OpenShift Cluster Manager, use subscription watch to track your OpenShift Container Platform subscriptions at the account or multi-cluster level. For more information about subscription watch, see "Data Gathered and Used by Red Hat's subscription services" in the *Additional resources* section.

Additional resources

- See [About remote health monitoring](#) for more information about the Telemetry service

3.3.18. Next steps

- [Customize your cluster](#).
- If necessary, you can [Remote health reporting](#).
- [Set up your registry and configure registry storage](#).
- Optional: [View the events from the vSphere Problem Detector Operator](#) to determine if the cluster has permission or storage configuration issues.
- Optional: if you created encrypted virtual machines, [create an encrypted storage class](#).

3.4. INSTALLING A CLUSTER ON VSphere WITH NETWORK CUSTOMIZATIONS

In OpenShift Container Platform version 4.19, you can install a cluster on VMware vSphere infrastructure that you provision with customized network configuration options. By customizing your network configuration, your cluster can coexist with existing IP address allocations in your environment and integrate with existing MTU and VXLAN configurations.

You must set most of the network configuration parameters during installation, and you can modify only **kubeProxy** configuration parameters in a running cluster.



IMPORTANT

The steps for performing a user-provisioned infrastructure installation are provided as an example only. Installing a cluster with infrastructure you provide requires knowledge of the vSphere platform and the installation process of OpenShift Container Platform. Use the user-provisioned infrastructure installation instructions as a guide; you are free to create the required resources through other methods.

3.4.1. Prerequisites

- You have completed the tasks in [Preparing to install a cluster using user-provisioned infrastructure](#).
- You reviewed your VMware platform licenses. Red Hat does not place any restrictions on your VMware licenses, but some VMware infrastructure components require licensing.
- You reviewed details about the [OpenShift Container Platform installation and update processes](#).
- You read the documentation on [selecting a cluster installation method and preparing it for users](#).
- Completing the installation requires that you upload the Red Hat Enterprise Linux CoreOS (RHCOS) OVA on vSphere hosts. The machine from which you complete this process requires access to port 443 on the vCenter and ESXi hosts. Verify that port 443 is accessible.
- If you use a firewall, you confirmed with the administrator that port 443 is accessible. Control plane nodes must be able to reach vCenter and ESXi hosts on port 443 for the installation to succeed.
- If you use a firewall, you [configured it to allow the sites](#) that your cluster requires access to.

3.4.2. Internet access for OpenShift Container Platform

In OpenShift Container Platform 4.19, you require access to the internet to install your cluster.

You must have internet access to perform the following actions:

- Access [OpenShift Cluster Manager](#) to download the installation program and perform subscription management. If the cluster has internet access and you do not disable Telemetry, that service automatically entitles your cluster.
- 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 required content and use it to populate a mirror registry with the installation packages. 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.

3.4.3. VMware vSphere region and zone enablement

You can deploy an OpenShift Container Platform cluster to multiple vSphere data centers. Each data center can run multiple clusters. This configuration reduces the risk of a hardware failure or network outage that can cause your cluster to fail. To enable regions and zones, you must define multiple failure domains for your OpenShift Container Platform cluster.



IMPORTANT

The VMware vSphere region and zone enablement feature requires the vSphere Container Storage Interface (CSI) driver as the default storage driver in the cluster. As a result, the feature is only available on a newly installed cluster.

For a cluster that was upgraded from a previous release, you must enable CSI automatic migration for the cluster. You can then configure multiple regions and zones for the upgraded cluster.

The default installation configuration deploys a cluster to a single vSphere data center. If you want to deploy a cluster to multiple vSphere data centers, you must create an installation configuration file that enables the region and zone feature.

The default **install-config.yaml** file includes **vcenters** and **failureDomains** fields, where you can specify multiple vSphere data centers and clusters for your OpenShift Container Platform cluster. You can leave these fields blank if you want to install an OpenShift Container Platform cluster in a vSphere environment that consists of single data center.

The following list describes terms associated with defining zones and regions for your cluster:

- Failure domain: Establishes the relationships between a region and zone. You define a failure domain by using vCenter objects, such as a **datastore** object. A failure domain defines the vCenter location for OpenShift Container Platform cluster nodes.
- Region: Specifies a vCenter data center. You define a region by using a tag from the **openshift-region** tag category.
- Zone: Specifies a vCenter cluster. You define a zone by using a tag from the **openshift-zone** tag category.



NOTE

If you plan on specifying more than one failure domain in your **install-config.yaml** file, you must create tag categories, zone tags, and region tags in advance of creating the configuration file.

You must create a vCenter tag for each vCenter data center, which represents a region. Additionally, you must create a vCenter tag for each cluster than runs in a data center, which represents a zone. After you create the tags, you must attach each tag to their respective data centers and clusters.

The following table outlines an example of the relationship among regions, zones, and tags for a configuration with multiple vSphere data centers running in a single VMware vCenter.

Data center (region)	Cluster (zone)	Tags
us-east	us-east-1	us-east-1a
		us-east-1b
	us-east-2	us-east-2a
		us-east-2b
us-west	us-west-1	us-west-1a
		us-west-1b
	us-west-2	us-west-2a
		us-west-2b

Additional resources

- [Additional VMware vSphere configuration parameters](#)
- [Deprecated VMware vSphere configuration parameters](#)
- [vSphere automatic migration](#)
- [VMware vSphere CSI Driver Operator](#)

3.4.4. Manually creating the installation configuration file

To customise your OpenShift Container Platform deployment and meet specific network requirements, manually create the installation configuration file. This ensures that the installation program uses your tailored settings rather than default values during the setup process.



IMPORTANT

The Cloud Controller Manager Operator performs a connectivity check on a provided hostname or IP address. Ensure that you specify a hostname or an IP address to a reachable vCenter server. If you provide metadata to a non-existent vCenter server, installation of the cluster fails at the bootstrap stage.

Prerequisites

- You have an SSH public key on your local machine for use with the installation program. You can use the key for SSH authentication onto your cluster nodes for debugging and disaster recovery.
- You have obtained the OpenShift Container Platform installation program and the pull secret for your cluster.

Procedure

1. Create an installation directory to store your required installation assets in:

```
$ mkdir <installation_directory>
```



IMPORTANT

You must create a directory. Some installation assets, such as 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.

2. Customize the provided sample **install-config.yaml** file template and save the file in the **<installation_directory>**.



NOTE

You must name this configuration file **install-config.yaml**.

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



IMPORTANT

Back up the **install-config.yaml** file now, because the installation process consumes the file in the next step.

Additional resources

- [Installation configuration parameters](#)

3.4.4.1. Sample **install-config.yaml** file for VMware vSphere

You can customize the **install-config.yaml** file to specify more details about your OpenShift Container Platform cluster's platform or modify the values of the required parameters.

```
additionalTrustBundlePolicy: Proxyonly
apiVersion: v1
baseDomain: example.com 1
compute: 2
- architecture: amd64
  name: <worker_node>
  platform: {}
```

```

replicas: 0 ③
controlPlane: ④
  architecture: amd64
  name: <parent_node>
  platform: {}
  replicas: 3 ⑤
metadata:
  creationTimestamp: null
  name: test ⑥
networking:
---
platform:
  vsphere:
    failureDomains: ⑦
    - name: <failure_domain_name>
      region: <default_region_name>
      server: <fully_qualified_domain_name>
    topology:
      computeCluster: "/<data_center>/host/<cluster>" ⑧
      datacenter: <data_center> ⑨
      datastore: "/<data_center>/datastore/<datastore>" ⑩
      networks:
        - <VM_Network_name>
      resourcePool: "/<data_center>/host/<cluster>/Resources/<resourcePool>" ⑪
      folder: "/<data_center_name>/vm/<folder_name>/<subfolder_name>" ⑫
    zone: <default_zone_name>
  vcenters:
    - datacenters:
      - <data_center>
        password: <password> ⑬
        port: 443
        server: <fully_qualified_domain_name> ⑭
        user: administrator@vsphere.local
        diskType: thin ⑮
    tips: false ⑯
    pullSecret: '{"auths": ...}' ⑰
    sshKey: 'ssh-ed25519 AAAA...' ⑱

```

- 1 The base domain of the cluster. All DNS records must be sub-domains of this base and include the cluster name.
- 2 ④ The **controlPlane** section is a single mapping, but the **compute** section is a sequence of mappings. To meet the requirements of the different data structures, the first line of the **compute** section must begin with a hyphen, -, and the first line of the **controlPlane** section must not. Both sections define a single machine pool, so only one control plane is used. OpenShift Container Platform does not support defining multiple compute pools.
- 3 You must set the value of the **replicas** parameter to **0**. This parameter controls the number of workers that the cluster creates and manages for you, which are functions that the cluster does not perform when you use user-provisioned infrastructure. You must manually deploy worker machines for the cluster to use before you finish installing OpenShift Container Platform.
- 5

The number of control plane machines that you add to the cluster. Because the cluster uses this value as the number of etcd endpoints in the cluster, the value must match the number of control

- 6 The cluster name that you specified in your DNS records.
- 7 Establishes the relationships between a region and zone. You define a failure domain by using vCenter objects, such as a **datastore** object. A failure domain defines the vCenter location for OpenShift Container Platform cluster nodes.
- 8 The vSphere data center.
- 9 The path to the vSphere datastore that holds virtual machine files, templates, and ISO images.

IMPORTANT

You can specify the path of any datastore that exists in a datastore cluster. By default, Storage vMotion is automatically enabled for a datastore cluster. Red Hat does not support Storage vMotion, so you must disable Storage vMotion to avoid data loss issues for your OpenShift Container Platform cluster.

If you must specify VMs across multiple datastores, use a **datastore** object to specify a failure domain in your cluster's **install-config.yaml** configuration file. For more information, see "VMware vSphere region and zone enablement".

- 10 Optional: For installer-provisioned infrastructure, the absolute path of an existing resource pool where the installation program creates the virtual machines, for example, `/<data_center_name>/host/<cluster_name>/Resources/<resource_pool_name>/<optional_named_resource_pool_name>`. If you do not specify a value, resources are installed in the root of the cluster `/example_data_center/host/example_cluster/Resources`.
- 11 Optional: For installer-provisioned infrastructure, the absolute path of an existing folder where the installation program creates the virtual machines, for example, `/<data_center_name>/vm/<folder_name>/<subfolder_name>`. If you do not provide this value, the installation program creates a top-level folder in the data center virtual machine folder that is named with the infrastructure ID. If you are providing the infrastructure for the cluster and you do not want to use the default **StorageClass** object, named **thin**, you can omit the **folder** parameter from the **install-config.yaml** file.
- 12 The password associated with the vSphere user.
- 13 The fully-qualified hostname or IP address of the vCenter server.

IMPORTANT

The Cloud Controller Manager Operator performs a connectivity check on a provided hostname or IP address. Ensure that you specify a hostname or an IP address to a reachable vCenter server. If you provide metadata to a non-existent vCenter server, installation of the cluster fails at the bootstrap stage.

- 14 The vSphere disk provisioning method.
- 15 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.



IMPORTANT

To enable FIPS mode for your cluster, you must run the installation program from a Red Hat Enterprise Linux (RHEL) computer configured to operate in FIPS mode. For more information about configuring FIPS mode on RHEL, see [Switching RHEL to FIPS mode](#).

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.

- 16 The pull secret that you obtained from [OpenShift Cluster Manager](#). 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.
- 17 The public portion of the default SSH key for the **core** user in Red Hat Enterprise Linux CoreOS (RHCOS).

3.4.4.2. Configuring the cluster-wide proxy during installation

To enable internet access in environments that deny direct connections, configure a cluster-wide proxy in the **install-config.yaml** file. This configuration ensures that the new OpenShift Container Platform cluster routes traffic through the specified HTTP or HTTPS proxy.

Prerequisites

- You have an existing **install-config.yaml** file.
- You have reviewed the sites that your cluster requires access to and determined whether any of them need to bypass the proxy. By default, all cluster egress traffic is proxied, including calls to hosting cloud provider APIs. You added sites to the **Proxy** object's **spec.noProxy** field to bypass the proxy if necessary.



NOTE

The **Proxy** object **status.noProxy** field is populated with the values of the **networking.machineNetwork[].cidr**, **networking.clusterNetwork[].cidr**, and **networking.serviceNetwork[]** fields from your installation configuration.

For installations on Amazon Web Services (AWS), Google Cloud, Microsoft Azure, and Red Hat OpenStack Platform (RHOSP), the **Proxy** object **status.noProxy** field is also populated with the instance metadata endpoint (**169.254.169.254**).

Procedure

1. Edit your **install-config.yaml** file and add the proxy settings. For example:

```
apiVersion: v1
baseDomain: my.domain.com
proxy:
```

```

httpProxy: http://<username>:<pswd>@<ip>:<port>
httpsProxy: https://<username>:<pswd>@<ip>:<port>
noProxy: example.com
additionalTrustBundle: |
  -----BEGIN CERTIFICATE-----
  <MY_TRUSTED_CA_CERT>
  -----END CERTIFICATE-----
additionalTrustBundlePolicy: <policy_to_add_additionalTrustBundle>
# ...

```

where:

proxy.httpProxy

Specifies a proxy URL to use for creating HTTP connections outside the cluster. The URL scheme must be **http**.

proxy.httpsProxy

Specifies a proxy URL to use for creating HTTPS connections outside the cluster.

proxy.noProxy

Specifies a comma-separated list of destination domain names, IP addresses, or other network CIDRs to exclude from proxying. Preface a domain with `.` to match subdomains only. For example, `.y.com` matches `x.y.com`, but not `y.com`. Use `*` to bypass the proxy for all destinations. You must include vCenter’s IP address and the IP range that you use for its machines.

additionalTrustBundle

If provided, the installation program generates a config map that is named **user-ca-bundle** in the **openshift-config** namespace to hold the additional CA certificates. If you provide **additionalTrustBundle** and at least one proxy setting, the **Proxy** object is configured to reference the **user-ca-bundle** config map in the **trustedCA** field. The Cluster Network Operator then creates a **trusted-ca-bundle** config map that merges the contents specified for the **trustedCA** parameter with the RHCOS trust bundle. The **additionalTrustBundle** field is required unless the proxy’s identity certificate is signed by an authority from the RHCOS trust bundle.

additionalTrustBundlePolicy

Specifies the policy that determines the configuration of the **Proxy** object to reference the **user-ca-bundle** config map in the **trustedCA** field. The allowed values are **Proxyonly** and **Always**. Use **Proxyonly** to reference the **user-ca-bundle** config map only when **http/https** proxy is configured. Use **Always** to always reference the **user-ca-bundle** config map. The default value is **Proxyonly**. Optional parameter.



NOTE

The installation program does not support the proxy **readinessEndpoints** field.

**NOTE**

If the installer times out, restart and then complete the deployment by using the **wait-for** command of the installer. For example:

+

```
$ ./openshift-install wait-for install-complete --log-level debug
```

2. Save the file and reference it when installing OpenShift Container Platform.

The installation program creates a cluster-wide proxy that is named **cluster** that uses the proxy settings in the provided **install-config.yaml** file. If no proxy settings are provided, a **cluster Proxy** object is still created, but it will have a nil **spec**.

**NOTE**

Only the **Proxy** object named **cluster** is supported, and no additional proxies can be created.

3.4.4.3. Configuring regions and zones for a VMware vCenter

You can modify the default installation configuration file, so that you can deploy an OpenShift Container Platform cluster to multiple vSphere data centers.

The default **install-config.yaml** file configuration from the previous release of OpenShift Container Platform is deprecated. You can continue to use the deprecated default configuration, but the **openshift-installer** will prompt you with a warning message that indicates the use of deprecated fields in the configuration file.

Prerequisites

- You have an existing **install-config.yaml** installation configuration file.

**IMPORTANT**

You must specify at least one failure domain for your OpenShift Container Platform cluster, so that you can provision data center objects for your VMware vCenter server. Consider specifying multiple failure domains if you need to provision virtual machine nodes in different data centers, clusters, datastores, and other components. To enable regions and zones, you must define multiple failure domains for your OpenShift Container Platform cluster.

- You have installed the **govc** command line tool.

**IMPORTANT**

The example uses the **govc** command. The **govc** command is an open source command available from VMware; it is not available from Red Hat. The Red Hat support team does not maintain the **govc** command. Instructions for downloading and installing **govc** are found on the VMware documentation website.

Procedure

1. Create the **openshift-region** and **openshift-zone** vCenter tag categories by running the following commands:



IMPORTANT

If you specify different names for the **openshift-region** and **openshift-zone** vCenter tag categories, the installation of the OpenShift Container Platform cluster fails.

```
$ govc tags.category.create -d "OpenShift region" openshift-region
```

```
$ govc tags.category.create -d "OpenShift zone" openshift-zone
```

2. For each region where you want to deploy your cluster, create a region tag by running the following command:

```
$ govc tags.create -c <region_tag_category> <region_tag>
```

3. For each zone where you want to deploy your cluster, create a zone tag by running the following command:

```
$ govc tags.create -c <zone_tag_category> <zone_tag>
```

4. Attach region tags to each vCenter data center object by running the following command:

```
$ govc tags.attach -c <region_tag_category> <region_tag_1> /<data_center_1>
```

5. Attach the zone tags to each vCenter cluster object by running the following command:

```
$ govc tags.attach -c <zone_tag_category> <zone_tag_1> /<data_center_1>/host/<cluster1>
```

6. Change to the directory that contains the installation program and initialize the cluster deployment according to your chosen installation requirements.

Sample `install-config.yaml` file with multiple data centers defined in a vSphere center

```
# ...
compute:
---
vsphere:
  zones:
    - "<machine_pool_zone_1>"
    - "<machine_pool_zone_2>"
# ...
controlPlane:
# ...
vsphere:
  zones:
    - "<machine_pool_zone_1>"
    - "<machine_pool_zone_2>"
```

```

# ...
platform:
  vsphere:
    vcenters:
# ...
  datacenters:
    - <data_center_1_name>
    - <data_center_2_name>
  failureDomains:
    - name: <machine_pool_zone_1>
      region: <region_tag_1>
      zone: <zone_tag_1>
      server: <fully_qualified_domain_name>
      topology:
        datacenter: <data_center_1>
        computeCluster: "/<data_center_1>/host/<cluster1>""
        networks:
          - <VM_Network1_name>
          datastore: "/<data_center_1>/datastore/<datastore1>""
          resourcePool: "/<data_center_1>/host/<cluster1>/Resources/<resourcePool1>""
          folder: "/<data_center_1>/vm/<folder1>""
    - name: <machine_pool_zone_2>
      region: <region_tag_2>
      zone: <zone_tag_2>
      server: <fully_qualified_domain_name>
      topology:
        datacenter: <data_center_2>
        computeCluster: "/<data_center_2>/host/<cluster2>""
        networks:
          - <VM_Network2_name>
          datastore: "/<data_center_2>/datastore/<datastore2>""
          resourcePool: "/<data_center_2>/host/<cluster2>/Resources/<resourcePool2>""
          folder: "/<data_center_2>/vm/<folder2>""
# ...

```

3.4.5. Network configuration phases

There are two phases prior to OpenShift Container Platform installation where you can customize the network configuration.

Phase 1

You can customize the following network-related fields in the **install-config.yaml** file before you create the manifest files:

- **networking.networkType**
- **networking.clusterNetwork**
- **networking.serviceNetwork**
- **networking.machineNetwork**
- **nodeNetworking**

For more information, see "Installation configuration parameters".



NOTE

Set the **networking.machineNetwork** to match the Classless Inter-Domain Routing (CIDR) where the preferred subnet is located.



IMPORTANT

The CIDR range **172.17.0.0/16** is reserved by **libVirt**. You cannot use any other CIDR range that overlaps with the **172.17.0.0/16** CIDR range for networks in your cluster.

Phase 2

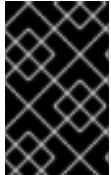
After creating the manifest files by running **openshift-install create manifests**, you can define a customized Cluster Network Operator manifest with only the fields you want to modify. You can use the manifest to specify an advanced network configuration.

During phase 2, you cannot override the values that you specified in phase 1 in the **install-config.yaml** file. However, you can customize the network plugin during phase 2.

3.4.6. Specifying advanced network configuration

You can use advanced network configuration for your network plugin to integrate your cluster into your existing network environment.

You can specify advanced network configuration only before you install the cluster.



IMPORTANT

Customizing your network configuration by modifying the OpenShift Container Platform manifest files created by the installation program is not supported. Applying a manifest file that you create, as in the following procedure, is supported.

Prerequisites

- You have created the **install-config.yaml** file and completed any modifications to it.

Procedure

1. Change to the directory that contains the installation program and create the manifests:

\$./openshift-install create manifests --dir <installation_directory> 1

1 **<installation_directory>** specifies the name of the directory that contains the **install-config.yaml** file for your cluster.

2. Create a stub manifest file for the advanced network configuration that is named **cluster-network-03-config.yaml** in the **<installation_directory>/manifests** directory:

```
apiVersion: operator.openshift.io/v1
kind: Network
metadata:
```

```

  name: cluster
  spec:

```

3. Specify the advanced network configuration for your cluster in the **cluster-network-03-config.yml** file, such as in the following example:

Enable IPsec for the OVN-Kubernetes network provider

```

apiVersion: operator.openshift.io/v1
kind: Network
metadata:
  name: cluster
spec:
  defaultNetwork:
    ovnKubernetesConfig:
      ipsecConfig:
        mode: Full

```

4. Optional: Back up the **manifests/cluster-network-03-config.yml** file. The installation program consumes the **manifests** directory when you create the Ignition config files.
5. Remove the Kubernetes manifest files that define the control plane machines and compute **MachineSets**:

```
$ rm -f openshift/99_openshift-cluster-api_master-machines-*.yaml openshift/99_openshift-cluster-api_worker-machineset-*.yaml
```

Because you create and manage these resources yourself, you do not have to initialize them.

- You can preserve the **MachineSet** files to create compute machines by using the machine API, but you must update references to them to match your environment.

3.4.6.1. Specifying multiple subnets for your network

Before you install an OpenShift Container Platform cluster on a vSphere host, you can specify multiple subnets for a networking implementation so that the vSphere cloud controller manager (CCM) can select the appropriate subnet for a given networking situation. vSphere can use the subnet for managing pods and services on your cluster.

For this configuration, you must specify internal and external Classless Inter-Domain Routing (CIDR) implementations in the vSphere CCM configuration. Each CIDR implementation lists an IP address range that the CCM uses to decide what subnets interact with traffic from internal and external networks.



IMPORTANT

Failure to configure internal and external CIDR implementations in the vSphere CCM configuration can cause the vSphere CCM to select the wrong subnet. This situation causes the following error:

ERROR Bootstrap failed to complete: timed out waiting for the condition
 ERROR Failed to wait for bootstrapping to complete. This error usually happens when there is a problem with control plane hosts that prevents the control plane operators from creating the control plane.

This configuration can cause new nodes that associate with a **MachineSet** object with a single subnet to become unusable as each new node receives the **node.cloudprovider.kubernetes.io/uninitialized** taint. These situations can cause communication issues with the Kubernetes API server that can cause installation of the cluster to fail.

Prerequisites

- You created Kubernetes manifest files for your OpenShift Container Platform cluster.

Procedure

1. From the directory where you store your OpenShift Container Platform cluster manifest files, open the **manifests/cluster-infrastructure-02-config.yml** manifest file.
2. Add a **nodeNetworking** object to the file and specify internal and external network subnet CIDR implementations for the object.

TIP

For most networking situations, consider setting the standard multiple-subnet configuration. This configuration requires that you set the same IP address ranges in the **nodeNetworking.internal.networkSubnetCidr** and **nodeNetworking.external.networkSubnetCidr** parameters.

Example of a configured **cluster-infrastructure-02-config.yml** manifest file

```
apiVersion: config.openshift.io/v1
kind: Infrastructure
metadata:
  name: cluster
spec:
  cloudConfig:
    key: config
    name: cloud-provider-config
  platformSpec:
    type: VSphere
  vsphere:
    failureDomains:
      - name: generated-failure-domain
    ...
  nodeNetworking:
    external:
```

```

networkSubnetCidr:
- <machine_network_cidr_ipv4>
- <machine_network_cidr_ipv6>
internal:
networkSubnetCidr:
- <machine_network_cidr_ipv4>
- <machine_network_cidr_ipv6>
# ...

```

Additional resources

- [Cluster Network Operator configuration](#)
- [.spec.platformSpec.vsphere.nodeNetworking](#)

3.4.7. Cluster Network Operator configuration

To manage cluster networking, configure the Cluster Network Operator (CNO) **Network** custom resource (CR) named **cluster** so the cluster uses the correct IP ranges and network plugin settings for reliable pod and service connectivity. Some settings and fields are inherited at the time of install or by the **default.Network.type** plugin, OVN-Kubernetes.

The CNO configuration inherits the following fields during cluster installation from the **Network** API in the **Network.config.openshift.io** API group:

clusterNetwork

IP address pools from which pod IP addresses are allocated.

serviceNetwork

IP address pool for services.

defaultNetwork.type

Cluster network plugin. **OVNKubernetes** is the only supported plugin during installation.

You can specify the cluster network plugin configuration for your cluster by setting the fields for the **defaultNetwork** object in the CNO object named **cluster**.

3.4.7.1. Cluster Network Operator configuration object

The fields for the Cluster Network Operator (CNO) are described in the following table:

Table 3.11. Cluster Network Operator configuration object

Field	Type	Description
metadata.name	string	The name of the CNO object. This name is always cluster .

Field	Type	Description
spec.clusterNetwork	array	<p>A list specifying the blocks of IP addresses from which pod IP addresses are allocated and the subnet prefix length assigned to each individual node in the cluster. For example:</p> <pre>spec: clusterNetwork: - cidr: 10.128.0.0/19 hostPrefix: 23 - cidr: 10.128.32.0/19 hostPrefix: 23</pre>
spec.serviceNetwork	array	<p>A block of IP addresses for services. The OVN-Kubernetes network plugin supports only a single IP address block for the service network. For example:</p> <pre>spec: serviceNetwork: - 172.30.0.0/14</pre> <p>You can customize this field only in the install-config.yaml file before you create the manifests. The value is read-only in the manifest file.</p>
spec.defaultNetwork	object	<p>Configures the network plugin for the cluster network.</p>
spec.additionalRoutingCapabilities.providers	array	<p>This setting enables a dynamic routing provider. The FRR routing capability provider is required for the route advertisement feature. The only supported value is FRR.</p> <ul style="list-style-type: none"> ● FRR: The FRR routing provider <pre>spec: additionalRoutingCapabilities: providers: - FRR</pre>



IMPORTANT

For a cluster that needs to deploy objects across multiple networks, ensure that you specify the same value for the **clusterNetwork.hostPrefix** parameter for each network type that is defined in the **install-config.yaml** file. Setting a different value for each **clusterNetwork.hostPrefix** parameter can impact the OVN-Kubernetes network plugin, where the plugin cannot effectively route object traffic among different nodes.

3.4.7.2. defaultNetwork object configuration

The values for the **defaultNetwork** object are defined in the following table:

Table 3.12. **defaultNetwork** object

Field	Type	Description
type	string	<p>OVNKubernetes. The Red Hat OpenShift Networking network plugin is selected during installation. This value cannot be changed after cluster installation.</p> <p> NOTE</p> <p>OpenShift Container Platform uses the OVN-Kubernetes network plugin by default.</p>
ovnKubernetesConfig	object	This object is only valid for the OVN-Kubernetes network plugin.

3.4.7.3. Configuration for the OVN-Kubernetes network plugin

The following table describes the configuration fields for the OVN-Kubernetes network plugin:

Table 3.13. **ovnKubernetesConfig** object

Field	Type	Description
mtu	integer	<p>The maximum transmission unit (MTU) for the Geneve (Generic Network Virtualization Encapsulation) overlay network. This is detected automatically based on the MTU of the primary network interface. You do not normally need to override the detected MTU.</p> <p>If the auto-detected value is not what you expect it to be, confirm that the MTU on the primary network interface on your nodes is correct. You cannot use this option to change the MTU value of the primary network interface on the nodes.</p> <p>If your cluster requires different MTU values for different nodes, you must set this value to 100 less than the lowest MTU value in your cluster. For example, if some nodes in your cluster have an MTU of 9001, and some have an MTU of 1500, you must set this value to 1400.</p>
genevePort	integer	The port to use for all Geneve packets. The default value is 6081 . This value cannot be changed after cluster installation.
ipsecConfig	object	Specify a configuration object for customizing the IPsec configuration.
ipv4	object	Specifies a configuration object for IPv4 settings.

Field	Type	Description
ipv6	object	Specifies a configuration object for IPv6 settings.
policyAuditConfig	object	Specify a configuration object for customizing network policy audit logging. If unset, the defaults audit log settings are used.
routeAdVERTISEMENTS	string	<p>Specifies whether to advertise cluster network routes. The default value is Disabled.</p> <ul style="list-style-type: none"> ● Enabled: Import routes to the cluster network and advertise cluster network routes as configured in RouteAdVERTISEMENTS objects. ● Disabled: Do not import routes to the cluster network or advertise cluster network routes.
gatewayConfig	object	<p>Optional: Specify a configuration object for customizing how egress traffic is sent to the node gateway. Valid values are Shared and Local. The default value is Shared. In the default setting, the Open vSwitch (OVS) outputs traffic directly to the node IP interface. In the Local setting, it traverses the host network; consequently, it gets applied to the routing table of the host.</p> <div style="display: flex; align-items: center;">  NOTE <p>While migrating egress traffic, you can expect some disruption to workloads and service traffic until the Cluster Network Operator (CNO) successfully rolls out the changes.</p> </div>

Table 3.14. `ovnKubernetesConfig.ipv4` object

Field	Type	Description
internalTransitSwitchSubnet	string	<p>If your existing network infrastructure overlaps with the 100.88.0.0/16 IPv4 subnet, you can specify a different IP address range for internal use by OVN-Kubernetes. The subnet for the distributed transit switch that enables east-west traffic. This subnet cannot overlap with any other subnets used by OVN-Kubernetes or on the host itself. It must be large enough to accommodate one IP address per node in your cluster.</p> <p>The default value is 100.88.0.0/16.</p>

Field	Type	Description
internalJoinSubnet	string	<p>If your existing network infrastructure overlaps with the 100.64.0.0/16 IPv4 subnet, you can specify a different IP address range for internal use by OVN-Kubernetes. You must ensure that the IP address range does not overlap with any other subnet used by your OpenShift Container Platform installation. The IP address range must be larger than the maximum number of nodes that can be added to the cluster. For example, if the clusterNetwork.cidr value is 10.128.0.0/14 and the clusterNetwork.hostPrefix value is /23, then the maximum number of nodes is 2^(23-14)=512.</p> <p>The default value is 100.64.0.0/16.</p>

Table 3.15. **ovnKubernetesConfig.ipv6** object

Field	Type	Description
internalTransitSwitchSubnet	string	<p>If your existing network infrastructure overlaps with the fd97::/64 IPv6 subnet, you can specify a different IP address range for internal use by OVN-Kubernetes. The subnet for the distributed transit switch that enables east-west traffic. This subnet cannot overlap with any other subnets used by OVN-Kubernetes or on the host itself. It must be large enough to accommodate one IP address per node in your cluster.</p> <p>The default value is fd97::/64.</p>
internalJoinSubnet	string	<p>If your existing network infrastructure overlaps with the fd98::/64 IPv6 subnet, you can specify a different IP address range for internal use by OVN-Kubernetes. You must ensure that the IP address range does not overlap with any other subnet used by your OpenShift Container Platform installation. The IP address range must be larger than the maximum number of nodes that can be added to the cluster.</p> <p>The default value is fd98::/64.</p>

Table 3.16. **policyAuditConfig** object

Field	Type	Description
rateLimit	integer	The maximum number of messages to generate every second per node. The default value is 20 messages per second.
maxFileSize	integer	The maximum size for the audit log in bytes. The default value is 50000000 or 50 MB.

Field	Type	Description
maxLogFiles	integer	The maximum number of log files that are retained.
destination	string	<p>One of the following additional audit log targets:</p> <p>libc The libc syslog() function of the journald process on the host.</p> <p>udp:<host>:<port> A syslog server. Replace <host>:<port> with the host and port of the syslog server.</p> <p>unix:<file> A Unix Domain Socket file specified by <file>.</p> <p>null Do not send the audit logs to any additional target.</p>
syslogFacility	string	The syslog facility, such as kern , as defined by RFC5424. The default value is local0 .

Table 3.17. **gatewayConfig** object

Field	Type	Description
routingViaHost	boolean	<p>Set this field to true to send egress traffic from pods to the host networking stack. For highly-specialized installations and applications that rely on manually configured routes in the kernel routing table, you might want to route egress traffic to the host networking stack. By default, egress traffic is processed in OVN to exit the cluster and is not affected by specialized routes in the kernel routing table. The default value is false.</p> <p>This field has an interaction with the Open vSwitch hardware offloading feature. If you set this field to true, you do not receive the performance benefits of the offloading because egress traffic is processed by the host networking stack.</p>
ipForwarding	object	<p>You can control IP forwarding for all traffic on OVN-Kubernetes managed interfaces by using the ipForwarding specification in the Network resource. Specify Restricted to only allow IP forwarding for Kubernetes related traffic. Specify Global to allow forwarding of all IP traffic. For new installations, the default is Restricted. For updates to OpenShift Container Platform 4.14 or later, the default is Global.</p> <div style="display: flex; align-items: center;">  <div style="margin-left: 10px;"> <p>NOTE</p> <p>The default value of Restricted sets the IP forwarding to drop.</p> </div> </div>

Field	Type	Description
ipv4	object	Optional: Specify an object to configure the internal OVN-Kubernetes masquerade address for host to service traffic for IPv4 addresses.
ipv6	object	Optional: Specify an object to configure the internal OVN-Kubernetes masquerade address for host to service traffic for IPv6 addresses.

Table 3.18. `gatewayConfig.ipv4` object

Field	Type	Description
internalMasqueradeSubnet	string	<p>The masquerade IPv4 addresses that are used internally to enable host to service traffic. The host is configured with these IP addresses as well as the shared gateway bridge interface. The default value is 169.254.169.0/29.</p> <div style="display: flex; align-items: center; justify-content: space-between;"> <div style="flex: 1; text-align: center;">  </div> <div> IMPORTANT <p>For OpenShift Container Platform 4.17 and later versions, clusters use 169.254.0.0/17 as the default masquerade subnet. For upgraded clusters, there is no change to the default masquerade subnet.</p> </div> </div>

Table 3.19. `gatewayConfig.ipv6` object

Field	Type	Description
internalMasqueradeSubnet	string	<p>The masquerade IPv6 addresses that are used internally to enable host to service traffic. The host is configured with these IP addresses as well as the shared gateway bridge interface. The default value is fd69::/125.</p> <div style="display: flex; align-items: center; justify-content: space-between;"> <div style="flex: 1; text-align: center;">  </div> <div> IMPORTANT <p>For OpenShift Container Platform 4.17 and later versions, clusters use fd69::/112 as the default masquerade subnet. For upgraded clusters, there is no change to the default masquerade subnet.</p> </div> </div>

Table 3.20. `ipsecConfig` object

Field	Type	Description
mode	string	Specifies the behavior of the IPsec implementation. Must be one of the following values: <ul style="list-style-type: none"> • Disabled: IPsec is not enabled on cluster nodes. • External: IPsec is enabled for network traffic with external hosts. • Full: IPsec is enabled for pod traffic and network traffic with external hosts.

Example OVN-Kubernetes configuration with IPsec enabled

```
defaultNetwork:
type: OVNKubernetes
ovnKubernetesConfig:
  mtu: 1400
  genevePort: 6081
  ipsecConfig:
    mode: Full
```

3.4.8. Creating the Ignition config files

Because you must manually start the cluster machines, you must generate the Ignition config files that the cluster needs to make its machines.



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.
- It is recommended that you use 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.

Prerequisites

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

Procedure

- Obtain the Ignition config files:

```
$ ./openshift-install create ignition-configs --dir <installation_directory> 1
```

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



IMPORTANT

If you created an **install-config.yaml** file, specify the directory that contains it. Otherwise, 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.

The following files are generated in the directory:

```
.
  └── auth
      ├── kubeadmin-password
      └── kubeconfig
  └── bootstrap.ign
  └── master.ign
  └── metadata.json
  └── worker.ign
```

3.4.9. Extracting the infrastructure name

The Ignition config files contain a unique cluster identifier that you can use to uniquely identify your cluster in VMware vSphere. If you plan to use the cluster identifier as the name of your virtual machine folder, you must extract it.

Prerequisites

- You obtained the OpenShift Container Platform installation program and the pull secret for your cluster.
- You generated the Ignition config files for your cluster.
- You installed the **jq** package.

Procedure

- To extract and view the infrastructure name from the Ignition config file metadata, run the following command:

```
$ jq -r .infraID <installation_directory>/metadata.json 1
```

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

Example output

```
openshift-vw9j6 1
```

1 The output of this command is your cluster name and a random string.

3.4.10. Installing RHCOS and starting the OpenShift Container Platform bootstrap process

To install OpenShift Container Platform on user-provisioned infrastructure on VMware vSphere, you must install Red Hat Enterprise Linux CoreOS (RHCOS) on vSphere hosts. When you install RHCOS, you must provide the Ignition config file that was generated by the OpenShift Container Platform installation program for the type of machine you are installing. If you have configured suitable networking, DNS, and load balancing infrastructure, the OpenShift Container Platform bootstrap process begins automatically after the RHCOS machines have rebooted.

Prerequisites

- You have obtained the Ignition config files for your cluster.
- You have access to an HTTP server that you can access from your computer and that the machines that you create can access.
- You have created a [vSphere cluster](#).

Procedure

- 1 Upload the bootstrap Ignition config file, which is named **<installation_directory>/bootstrap.ign**, that the installation program created to your HTTP server. Note the URL of this file.
- 2 Save the following secondary Ignition config file for your bootstrap node to your computer as **<installation_directory>/merge-bootstrap.ign**:

```
{
  "ignition": {
    "config": {
      "merge": [
        {
          "source": "<bootstrap_ignition_config_url>", 1
          "verification": {}
        }
      ]
    },
    "timeouts": {},
    "version": "3.2.0"
  },
  "networkd": {},
  "passwd": {},
  "storage": {},
  "systemd": {}
}
```

- 1 Specify the URL of the bootstrap Ignition config file that you hosted.

When you create the virtual machine (VM) for the bootstrap machine, you use this Ignition config file.

- 3 Locate the following Ignition config files that the installation program created:

- **<installation_directory>/master.ign**
- **<installation_directory>/worker.ign**
- **<installation_directory>/merge-bootstrap.ign**

- 4 Convert the Ignition config files to Base64 encoding. Later in this procedure, you must add these files to the extra configuration parameter **guestinfo.ignition.config.data** in your VM. For example, if you use a Linux operating system, you can use the **base64** command to encode the files.

```
$ base64 -w0 <installation_directory>/master.ign > <installation_directory>/master.64
```

```
$ base64 -w0 <installation_directory>/worker.ign > <installation_directory>/worker.64
```

```
$ base64 -w0 <installation_directory>/merge-bootstrap.ign > <installation_directory>/merge-bootstrap.64
```



IMPORTANT

If you plan to add more compute machines to your cluster after you finish installation, do not delete these files.

- 5 Obtain the RHCOS OVA image. Images are available from the [RHCOS image mirror](#) page.



IMPORTANT

The RHCOS images might not change with every release of OpenShift Container Platform. You must download an image with the highest version that is less than or equal to the OpenShift Container Platform version that you install. Use the image version that matches your OpenShift Container Platform version if it is available.

The filename contains the OpenShift Container Platform version number in the format **rhcose-vmware.<architecture>.ova**.

- 6 In the vSphere Client, create a folder in your data center to store your VMs.
 - a. Click the **VMs and Templates** view.
 - b. Right-click the name of your data center.
 - c. Click **New Folder → New VM and Template Folder**.
 - d. In the window that is displayed, enter the folder name. If you did not specify an existing folder in the **install-config.yaml** file, then create a folder with the same name as the

infrastructure ID. You use this folder name so vCenter dynamically provisions storage in the appropriate location for its Workspace configuration.

7. In the vSphere Client, create a template for the OVA image and then clone the template as needed.



NOTE

In the following steps, you create a template and then clone the template for all of your cluster machines. You then provide the location for the Ignition config file for that cloned machine type when you provision the VMs.

- From the **Hosts and Clusters** tab, right-click your cluster name and select **Deploy OVF Template**.
- On the **Select an OVF** tab, specify the name of the RHCOS OVA file that you downloaded.
- On the **Select a name and folder** tab, set a **Virtual machine name** for your template, such as **Template-RHCOS**. Click the name of your vSphere cluster and select the folder you created in the previous step.
- On the **Select a compute resource** tab, click the name of your vSphere cluster.
- On the **Select storage** tab, configure the storage options for your VM.
 - Select **Thin Provision** or **Thick Provision**, based on your storage preferences.
 - Select the datastore that you specified in your **install-config.yaml** file.
 - If you want to encrypt your virtual machines, select **Encrypt this virtual machine**. See the section titled "Requirements for encrypting virtual machines" for more information.
- On the **Select network** tab, specify the network that you configured for the cluster, if available.
- When creating the OVF template, do not specify values on the **Customize template** tab or configure the template any further.



IMPORTANT

Do not start the original VM template. The VM template must remain off and must be cloned for new RHCOS machines. Starting the VM template configures the VM template as a VM on the platform, which prevents it from being used as a template that compute machine sets can apply configurations to.

8. Optional: Update the configured virtual hardware version in the VM template, if necessary. Follow [Upgrading a virtual machine to the latest hardware version](#) in the VMware documentation for more information.



IMPORTANT

It is recommended that you update the hardware version of the VM template to version 15 before creating VMs from it, if necessary. Using hardware version 13 for your cluster nodes running on vSphere is now deprecated. If your imported template defaults to hardware version 13, you must ensure that your ESXi host is on 6.7U3 or later before upgrading the VM template to hardware version 15. If your vSphere version is less than 6.7U3, you can skip this upgrade step; however, a future version of OpenShift Container Platform is scheduled to remove support for hardware version 13 and vSphere versions less than 6.7U3.

9. After the template deploys, deploy a VM for a machine in the cluster.
 - a. Right-click the template name and click **Clone** → **Clone to Virtual Machine**
 - b. On the **Select a name and folder** tab, specify a name for the VM. You might include the machine type in the name, such as **control-plane-0** or **compute-1**.

NOTE

Ensure that all virtual machine names across a vSphere installation are unique.

- c. On the **Select a name and folder** tab, select the name of the folder that you created for the cluster.
- d. On the **Select a compute resource** tab, select the name of a host in your data center.
- e. On the **Select clone options** tab, select **Customize this virtual machine's hardware**.
- f. On the **Customize hardware** tab, click **Advanced Parameters**.



IMPORTANT

The following configuration suggestions are for example purposes only. As a cluster administrator, you must configure resources according to the resource demands placed on your cluster. To best manage cluster resources, consider creating a resource pool from the cluster's root resource pool.

- Optional: Override default DHCP networking in vSphere. To enable static IP networking:
 - Set your static IP configuration:

Example command

```
$ export IPCFG="ip=<ip>::<gateway>:<netmask>:<hostname>:<iface>:none  
nameserver=srv1 [nameserver=srv2 [nameserver=srv3 [...]]]"
```

Example command

```
$ export IPCFG="ip=192.168.100.101::192.168.100.254:255.255.255.0::none  
nameserver=8.8.8.8"
```

- Set the **guestinfo.afterburn.initrd.network-kargs** property before you boot a VM from an OVA in vSphere:

Example command

```
$ govc vm.change -vm "<vm_name>" -e "guestinfo.afterburn.initrd.network-kargs=${IPCFG}"
```

- Add the following configuration parameter names and values by specifying data in the **Attribute** and **Values** fields. Ensure that you select the **Add** button for each parameter that you create.
 - **guestinfo.ignition.config.data**: Locate the base-64 encoded files that you created previously in this procedure, and paste the contents of the base64-encoded Ignition config file for this machine type.
 - **guestinfo.ignition.config.data.encoding**: Specify **base64**.
 - **disk.EnableUUID**: Specify **TRUE**.
 - **stealclock.enable**: If this parameter was not defined, add it and specify **TRUE**.
 - Create a child resource pool from the cluster's root resource pool. Perform resource allocation in this child resource pool.
- g. In the **Virtual Hardware** panel of the **Customize hardware** tab, modify the specified values as required. Ensure that the amount of RAM, CPU, and disk storage meets the minimum requirements for the machine type.
- h. Complete the remaining configuration steps. On clicking the **Finish** button, you have completed the cloning operation.
 - i. From the **Virtual Machines** tab, right-click on your VM and then select **Power → Power On**.
 - j. Check the console output to verify that Ignition ran.

Example command

```
Ignition: ran on 2022/03/14 14:48:33 UTC (this boot)
Ignition: user-provided config was applied
```

Next steps

- Create the rest of the machines for your cluster by following the preceding steps for each machine.



IMPORTANT

You must create the bootstrap and control plane machines at this time. Because some pods are deployed on compute machines by default, also create at least two compute machines before you install the cluster.

3.4.11. Adding more compute machines to a cluster in vSphere

You can add more compute machines to a user-provisioned OpenShift Container Platform cluster on VMware vSphere.

After your vSphere template deploys in your OpenShift Container Platform cluster, you can deploy a virtual machine (VM) for a machine in that cluster.

Prerequisites

- Obtain the base64-encoded Ignition file for your compute machines.
- You have access to the vSphere template that you created for your cluster.

Procedure

1. Right-click the template's name and click **Clone** → **Clone to Virtual Machine**
2. On the **Select a name and folder** tab, specify a name for the VM. You might include the machine type in the name, such as **compute-1**.
3. On the **Select a name and folder** tab, select the name of the folder that you created for the cluster.
4. On the **Select a compute resource** tab, select the name of a host in your data center.
5. On the **Select storage** tab, select storage for your configuration and disk files.
6. On the **Select clone options** tab, select **Customize this virtual machine's hardware**
7. On the **Customize hardware** tab, click **Advanced Parameters**.
 - Add the following configuration parameter names and values by specifying data in the **Attribute** and **Values** fields. Ensure that you select the **Add** button for each parameter that you create.
 - **guestinfo.ignition.config.data**: Paste the contents of the base64-encoded compute Ignition config file for this machine type.
 - **guestinfo.ignition.config.data.encoding**: Specify **base64**.
 - **disk.EnableUUID**: Specify **TRUE**.
8. In the **Virtual Hardware** panel of the **Customize hardware** tab, modify the specified values as required. Ensure that the amount of RAM, CPU, and disk storage meets the minimum requirements for the machine type. If many networks exist, select **Add New Device** > **Network Adapter**, and then enter your network information in the fields provided by the **New Network** menu item.
9. Complete the remaining configuration steps. On clicking the **Finish** button, you have completed the cloning operation.
10. From the **Virtual Machines** tab, right-click on your VM and then select **Power** → **Power On**.



NOTE

Ensure that all virtual machine names across a vSphere installation are unique.

Next steps

- Continue to create more compute machines for your cluster.

3.4.12. Disk partitioning

In most cases, data partitions are originally created by installing RHCOS, rather than by installing another operating system. In such cases, the OpenShift Container Platform installer should be allowed to configure your disk partitions.

However, there are two cases where you might want to intervene to override the default partitioning when installing an OpenShift Container Platform node:

- Create separate partitions: For greenfield installations on an empty disk, you might want to add separate storage to a partition. This is officially supported for making **/var** or a subdirectory of **/var**, such as **/var/lib/etcd**, a separate partition, but not both.



IMPORTANT

For disk sizes larger than 100GB, and especially disk sizes larger than 1TB, create a separate **/var** partition. See "Creating a separate **/var** partition" and this [Red Hat Knowledgebase article](#) for more information.



IMPORTANT

Kubernetes supports only two file system partitions. If you add more than one partition to the original configuration, Kubernetes cannot monitor all of them.

- Retain existing partitions: For a brownfield installation where you are reinstalling OpenShift Container Platform on an existing node and want to retain data partitions installed from your previous operating system, there are both boot arguments and options to **coreos-installer** that allow you to retain existing data partitions.

3.4.13. Creating a separate **/var** partition

In general, disk partitioning for OpenShift Container Platform should be left to the installer. However, there are cases where you might want to create separate partitions in a part of the filesystem that you expect to grow.

OpenShift Container Platform supports the addition of a single partition to attach storage to either the **/var** partition or a subdirectory of **/var**. For example:

- **/var/lib/containers**: Holds container-related content that can grow as more images and containers are added to a system.
- **/var/lib/etcd**: Holds data that you might want to keep separate for purposes such as performance optimization of etcd storage.
- **/var**: Holds data that you might want to keep separate for purposes such as auditing.



IMPORTANT

For disk sizes larger than 100GB, and especially larger than 1TB, create a separate **/var** partition.

Storing the contents of a **/var** directory separately makes it easier to grow storage for those areas as needed and reinstall OpenShift Container Platform at a later date and keep that data intact. With this method, you will not have to pull all your containers again, nor will you have to copy massive log files when you update systems.

Because **/var** must be in place before a fresh installation of Red Hat Enterprise Linux CoreOS (RHCOS), the following procedure sets up the separate **/var** partition by creating a machine config manifest that is inserted during the **openshift-install** preparation phases of an OpenShift Container Platform installation.

Procedure

1. Create a directory to hold the OpenShift Container Platform installation files:

```
$ mkdir $HOME/clusterconfig
```

2. Run **openshift-install** to create a set of files in the **manifest** and **openshift** subdirectories. Answer the system questions as you are prompted:

```
$ openshift-install create manifests --dir $HOME/clusterconfig
? SSH Public Key ...
$ ls $HOME/clusterconfig/openshift/
99_kubeadmin-password-secret.yaml
99_openshift-cluster-api_master-machines-0.yaml
99_openshift-cluster-api_master-machines-1.yaml
99_openshift-cluster-api_master-machines-2.yaml
...
```

3. Create a Butane config that configures the additional partition. For example, name the file **\$HOME/clusterconfig/98-var-partition.bu**, change the disk device name to the name of the storage device on the **worker** systems, and set the storage size as appropriate. This example places the **/var** directory on a separate partition:

```
variant: openshift
version: 4.19.0
metadata:
  labels:
    machineconfiguration.openshift.io/role: worker
  name: 98-var-partition
storage:
  disks:
    - device: /dev/disk/by-id/<device_name> ①
      partitions:
        - label: var
          start_mib: <partition_start_offset> ②
          size_mib: <partition_size> ③
          number: 5
  filesystems:
    - device: /dev/disk/by-partlabel/var
      path: /var
      format: xfs
      mount_options: [defaults, prjquota] ④
      with_mount_unit: true
```

- 1 The storage device name of the disk that you want to partition.
- 2 When adding a data partition to the boot disk, a minimum value of 25000 mebibytes is recommended. The root file system is automatically resized to fill all available space up to the specified offset. If no value is specified, or if the specified value is smaller than the recommended minimum, the resulting root file system will be too small, and future reinstalls of RHCOS might overwrite the beginning of the data partition.
- 3 The size of the data partition in mebibytes.
- 4 The **prjquota** mount option must be enabled for filesystems used for container storage.



NOTE

When creating a separate **/var** partition, you cannot use different instance types for worker nodes, if the different instance types do not have the same device name.

4. Create a manifest from the Butane config and save it to the **clusterconfig/openshift** directory. For example, run the following command:

```
$ butane $HOME/clusterconfig/98-var-partition.bu -o $HOME/clusterconfig/openshift/98-var-partition.yaml
```

5. Run **openshift-install** again to create Ignition configs from a set of files in the **manifest** and **openshift** subdirectories:

```
$ openshift-install create ignition-configs --dir $HOME/clusterconfig
$ ls $HOME/clusterconfig/
auth bootstrap.ign master.ign metadata.json worker.ign
```

Now you can use the Ignition config files as input to the vSphere installation procedures to install Red Hat Enterprise Linux CoreOS (RHCOS) systems.

3.4.14. Waiting for the bootstrap process to complete

To install OpenShift Container Platform, use Ignition configuration files to initialize the bootstrap process after the cluster nodes boot into RHCOS. You must wait for this process to complete to ensure the cluster is fully installed.

Prerequisites

- You have created the Ignition config files for your cluster.
- You have configured suitable network, DNS, and load balancing infrastructure.
- You have obtained the installation program and generated the Ignition config files for your cluster.
- You installed RHCOS on your cluster machines and provided the Ignition config files that the OpenShift Container Platform installation program generated.
- Your machines have direct internet access or have an HTTP or HTTPS proxy available.

Procedure

1. Monitor the bootstrap process:

```
$ ./openshift-install --dir <installation_directory> wait-for bootstrap-complete \
--log-level=info
```

where:

<installation_directory>

Specifies the path to the directory that stores the installation files.

--log-level=info

Specifies **warn**, **debug**, or **error** instead of **info** to view different installation details.

Example output

```
INFO Waiting up to 30m0s for the Kubernetes API at https://api.test.example.com:6443...
INFO API v1.32.3 up
INFO Waiting up to 30m0s for bootstrapping to complete...
INFO It is now safe to remove the bootstrap resources
```

The command succeeds when the Kubernetes API server signals that it has been bootstrapped on the control plane machines.

2. After the bootstrap process is complete, remove the bootstrap machine from the load balancer.



IMPORTANT

You must remove the bootstrap machine from the load balancer at this point. You can also remove or reformat the bootstrap machine itself.

3.4.15. Logging in to the cluster by using the CLI

To log in to your cluster as the default system user, export the **kubeconfig** file. This configuration enables the CLI to authenticate and connect to the specific API server created during OpenShift Container Platform installation.

The **kubeconfig** file is specific to a cluster and is created during OpenShift Container Platform installation.

Prerequisites

- You deployed an OpenShift Container Platform cluster.
- You installed the OpenShift CLI (**oc**).

Procedure

1. Export the **kubeadmin** credentials by running the following command:

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

where:

<installation_directory>

Specifies the path to the directory that stores the installation files.

- Verify you can run **oc** commands successfully using the exported configuration by running the following command:

```
$ oc whoami
```

Example output

```
system:admin
```

3.4.16. Approving the certificate signing requests for your machines

To add machines to a cluster, verify the status of the certificate signing requests (CSRs) generated for each machine. If manual approval is required, approve the client requests first, followed by the server requests.

Prerequisites

- You added machines to your cluster.

Procedure

- Confirm that the cluster recognizes the machines:

```
$ oc get nodes
```

Example output

```
NAME     STATUS   ROLES   AGE     VERSION
master-0  Ready    master   63m    v1.32.3
master-1  Ready    master   63m    v1.32.3
master-2  Ready    master   64m    v1.32.3
```

The output lists all of the machines that you created.



NOTE

The preceding output might not include the compute nodes, also known as worker nodes, until some CSRs are approved.

- Review the pending CSRs and ensure that you see the client requests with the **Pending** or **Approved** status for each machine that you added to the cluster:

```
$ oc get csr
```

Example output

NAME	AGE	REQUESTOR	CONDITION
csr-8b2br	15m	system:serviceaccount:openshift-machine-config-operator:node-bootstrapper	Pending
csr-8vnps	15m	system:serviceaccount:openshift-machine-config-operator:node-bootstrapper	Pending
...			

In this example, two machines are joining the cluster. You might see more approved CSRs in the list.

3. If the CSRs were not approved, after all of the pending CSRs for the machines you added are in **Pending** status, approve the CSRs for your cluster machines:



NOTE

Because the CSRs rotate automatically, approve your CSRs within an hour of adding the machines to the cluster. If you do not approve them within an hour, the certificates will rotate, and more than two certificates will be present for each node. You must approve all of these certificates. After the client CSR is approved, the Kubelet creates a secondary CSR for the serving certificate, which requires manual approval. Then, subsequent serving certificate renewal requests are automatically approved by the **machine-approver** if the Kubelet requests a new certificate with identical parameters.



NOTE

For clusters running on platforms that are not machine API enabled, such as bare metal and other user-provisioned infrastructure, you must implement a method of automatically approving the kubelet serving certificate requests (CSRs). If a request is not approved, then the **oc exec**, **oc rsh**, and **oc logs** commands cannot succeed, because a serving certificate is required when the API server connects to the kubelet. Any operation that contacts the Kubelet endpoint requires this certificate approval to be in place. The method must watch for new CSRs, confirm that the CSR was submitted by the **node-bootstrapper** service account in the **system:node** or **system:admin** groups, and confirm the identity of the node.

- To approve them individually, run the following command for each valid CSR:

```
$ oc adm certificate approve <csr_name>
```

where:

<csr_name>

Specifies the name of a CSR from the list of current CSRs.

- To approve all pending CSRs, run the following command:

```
$ oc get csr -o go-template='{{range .items}}{{if not .status}}{{.metadata.name}}\n{{end}}{{end}}' | xargs --no-run-if-empty oc adm certificate approve
```

**NOTE**

Some Operators might not become available until some CSRs are approved.

- Now that your client requests are approved, you must review the server requests for each machine that you added to the cluster:

```
$ oc get csr
```

Example output

NAME	AGE	REQUESTOR	CONDITION
csr-bfd72	5m26s	system:node:ip-10-0-50-126.us-east-2.compute.internal	Pending
csr-c571v	5m26s	system:node:ip-10-0-95-157.us-east-2.compute.internal	Pending
...			

- If the remaining CSRs are not approved, and are in the **Pending** status, approve the CSRs for your cluster machines:

- To approve them individually, run the following command for each valid CSR:

```
$ oc adm certificate approve <csr_name>
```

where:

<csr_name>

Specifies the name of a CSR from the list of current CSRs.

- To approve all pending CSRs, run the following command:

```
$ oc get csr -o go-template='{{range .items}}{{if not .status}}{{.metadata.name}}\n{{end}}{{end}}' | xargs oc adm certificate approve
```

- After all client and server CSRs have been approved, the machines have the **Ready** status. Verify this by running the following command:

```
$ oc get nodes
```

Example output

NAME	STATUS	ROLES	AGE	VERSION
master-0	Ready	master	73m	v1.32.3
master-1	Ready	master	73m	v1.32.3
master-2	Ready	master	74m	v1.32.3
worker-0	Ready	worker	11m	v1.32.3
worker-1	Ready	worker	11m	v1.32.3



NOTE

It can take a few minutes after approval of the server CSRs for the machines to transition to the **Ready** status.

3.4.16.1. Initial Operator configuration

To ensure all Operators become available, configure the required Operators immediately after the control plane initialises. This configuration is essential for stabilizing the cluster environment following the installation.

Prerequisites

- Your control plane has initialized.

Procedure

1. Watch the cluster components come online:

```
$ watch -n5 oc get clusteroperators
```

Example output

NAME SINCE	VERSION	AVAILABLE	PROGRESSING	DEGRADED
authentication	4.19.0	True	False	False 19m
baremetal	4.19.0	True	False	False 37m
cloud-credential	4.19.0	True	False	False 40m
cluster-autoscaler	4.19.0	True	False	False 37m
config-operator	4.19.0	True	False	False 38m
console	4.19.0	True	False	False 26m
csi-snapshot-controller	4.19.0	True	False	False 37m
dns	4.19.0	True	False	False 37m
etcd	4.19.0	True	False	False 36m
image-registry	4.19.0	True	False	False 31m
ingress	4.19.0	True	False	False 30m
insights	4.19.0	True	False	False 31m
kube-apiserver	4.19.0	True	False	False 26m
kube-controller-manager	4.19.0	True	False	False 36m
kube-scheduler	4.19.0	True	False	False 36m
kube-storage-version-migrator	4.19.0	True	False	False 37m
machine-api	4.19.0	True	False	False 29m
machine approver	4.19.0	True	False	False 37m
machine-config	4.19.0	True	False	False 36m
marketplace	4.19.0	True	False	False 37m
monitoring	4.19.0	True	False	False 29m
network	4.19.0	True	False	False 38m
node-tuning	4.19.0	True	False	False 37m
openshift-apiserver	4.19.0	True	False	False 32m
openshift-controller-manager	4.19.0	True	False	False 30m
openshift-samples	4.19.0	True	False	False 32m
operator-lifecycle-manager	4.19.0	True	False	False 37m
operator-lifecycle-manager-catalog	4.19.0	True	False	False 37m

operator-lifecycle-manager-packageserver	4.19.0	True	False	False	32m
service-ca	4.19.0	True	False	False	38m
storage	4.19.0	True	False	False	37m

- Configure the Operators that are not available.

3.4.16.2. Image registry removed during installation

On platforms that do not provide shareable object storage, the OpenShift Image Registry Operator bootstraps itself as **Removed**. This allows **openshift-installer** to complete installations on these platform types.

After installation, you must edit the Image Registry Operator configuration to switch the **managementState** from **Removed** to **Managed**. When this has completed, you must configure storage.

3.4.16.3. Image registry storage configuration

The Image Registry Operator is not initially available for platforms that do not provide default storage. After installation, you must configure your registry to use storage so that the Registry Operator is made available.

Configure a persistent volume, which is required for production clusters. Where applicable, you can configure an empty directory as the storage location for non-production clusters.

You can also allow the image registry to use block storage types by using the **Recreate** rollout strategy during upgrades.

3.4.16.3.1. Configuring block registry storage for VMware vSphere

To allow the image registry to use block storage types such as vSphere Virtual Machine Disk (VMDK) during upgrades as a cluster administrator, you can use the **Recreate** rollout strategy.



IMPORTANT

Block storage volumes are supported but not recommended for use with image registry on production clusters. An installation where the registry is configured on block storage is not highly available because the registry cannot have more than one replica.

Procedure

- Enter the following command to set the image registry storage as a block storage type, patch the registry so that it uses the **Recreate** rollout strategy, and runs with only **1** replica:

```
$ oc patch config.imageregistry.operator.openshift.io/cluster --type=merge -p '{"spec": {"rolloutStrategy": "Recreate", "replicas": 1}}'
```

- Provision the persistent volume (PV) for the block storage device, and create a persistent volume claim (PVC) for that volume. The requested block volume uses the `ReadWriteOnce` (RWO) access mode.

- Create a **pvc.yaml** file with the following contents to define a VMware vSphere **PersistentVolumeClaim** object:

```
kind: PersistentVolumeClaim
```

```

apiVersion: v1
metadata:
  name: image-registry-storage
  namespace: openshift-image-registry
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 100Gi

```

where:

name

Specifies a unique name that represents the **PersistentVolumeClaim** object.

namespace

Specifies the **namespace** for the **PersistentVolumeClaim** object, which is **openshift-image-registry**.

accessModes

Specifies the access mode of the persistent volume claim. With **ReadWriteOnce**, the volume can be mounted with read and write permissions by a single node.

storage

The size of the persistent volume claim.

- b. Enter the following command to create the **PersistentVolumeClaim** object from the file:

```
$ oc create -f pvc.yaml -n openshift-image-registry
```

3. Enter the following command to edit the registry configuration so that it references the correct PVC:

```
$ oc edit config.imageregistry.operator.openshift.io -o yaml
```

Example output

```

storage:
  pvc:
    claim:

```

By creating a custom PVC, you can leave the **claim** field blank for the default automatic creation of an **image-registry-storage** PVC.

For instructions about configuring registry storage so that it references the correct PVC, see [Configuring the registry for vSphere](#).

3.4.17. Completing installation on user-provisioned infrastructure

To finalize the installation on user-provisioned infrastructure, complete the cluster deployment after configuring the Operators. This ensures the cluster is fully operational on the infrastructure that you provide.

Prerequisites

- Your control plane has initialized.
- You have completed the initial Operator configuration.

Procedure

1. Confirm that all the cluster components are online with the following command:

```
$ watch -n5 oc get clusteroperators
```

Example output

NAME SINCE	VERSION	AVAILABLE	PROGRESSING	DEGRADED
authentication	4.19.0	True	False	False 19m
baremetal	4.19.0	True	False	False 37m
cloud-credential	4.19.0	True	False	False 40m
cluster-autoscaler	4.19.0	True	False	False 37m
config-operator	4.19.0	True	False	False 38m
console	4.19.0	True	False	False 26m
csi-snapshot-controller	4.19.0	True	False	False 37m
dns	4.19.0	True	False	False 37m
etcd	4.19.0	True	False	False 36m
image-registry	4.19.0	True	False	False 31m
ingress	4.19.0	True	False	False 30m
insights	4.19.0	True	False	False 31m
kube-apiserver	4.19.0	True	False	False 26m
kube-controller-manager	4.19.0	True	False	False 36m
kube-scheduler	4.19.0	True	False	False 36m
kube-storage-version-migrator	4.19.0	True	False	False 37m
machine-api	4.19.0	True	False	False 29m
machine-approver	4.19.0	True	False	False 37m
machine-config	4.19.0	True	False	False 36m
marketplace	4.19.0	True	False	False 37m
monitoring	4.19.0	True	False	False 29m
network	4.19.0	True	False	False 38m
node-tuning	4.19.0	True	False	False 37m
openshift-apiserver	4.19.0	True	False	False 32m
openshift-controller-manager	4.19.0	True	False	False 30m
openshift-samples	4.19.0	True	False	False 32m
operator-lifecycle-manager	4.19.0	True	False	False 37m
operator-lifecycle-manager-catalog	4.19.0	True	False	False 37m
operator-lifecycle-manager-packageserver	4.19.0	True	False	False 32m
service-ca	4.19.0	True	False	False 38m
storage	4.19.0	True	False	False 37m

Alternatively, the following command notifies you when all of the clusters are available. The command also retrieves and displays credentials:

```
$ ./openshift-install --dir <installation_directory> wait-for install-complete
```

where:

<installation_directory>

Specifies the path to the directory that you stored the installation files in.

Example output

INFO Waiting up to 30m0s for the cluster to initialize...

The command succeeds when the Cluster Version Operator finishes deploying the OpenShift Container Platform cluster from Kubernetes API server.

**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.
- It is recommended that you use 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.

2. Confirm that the Kubernetes API server is communicating with the pods.

a. To view a list of all pods, use the following command:

\$ oc get pods --all-namespaces

Example output

NAMESPACE	NAME	READY	STATUS
RESTARTS	AGE		
openshift-apiserver-operator	openshift-apiserver-operator-85cb746d55-zqhs8	1/1	
Running 1 9m			
openshift-apiserver	apiserver-67b9g	1/1	Running 0
3m			
openshift-apiserver	apiserver-ljcmx	1/1	Running 0
1m			
openshift-apiserver	apiserver-z25h4	1/1	Running 0
2m			
openshift-authentication-operator	authentication-operator-69d5d8bf84-vh2n8	1/1	
Running 0 5m			

b. View the logs for a pod that is listed in the output of the previous command by using the following command:

\$ oc logs <pod_name> -n <namespace>

■ where:

<namespace>

Specifies the pod name and namespace, as shown in the output of an earlier command. If the pod logs display, the Kubernetes API server can communicate with the cluster machines.

3. For an installation with Fibre Channel Protocol (FCP), additional steps are required to enable multipathing. Do not enable multipathing during installation.

See "Enabling multipathing with kernel arguments on RHCOS" in the *Postinstallation machine configuration tasks* documentation for more information.

You can add extra compute machines after the cluster installation is completed by following [Adding compute machines to vSphere](#).

3.4.18. Configuring vSphere DRS anti-affinity rules for control plane nodes

vSphere Distributed Resource Scheduler (DRS) anti-affinity rules can be configured to support higher availability of OpenShift Container Platform Control Plane nodes. Anti-affinity rules ensure that the vSphere Virtual Machines for the OpenShift Container Platform Control Plane nodes are not scheduled to the same vSphere Host.



IMPORTANT

- The following information applies to compute DRS only and does not apply to storage DRS.
- The **govc** command is an open-source command available from VMware; it is not available from Red Hat. The **govc** command is not supported by the Red Hat support.
- Instructions for downloading and installing **govc** are found on the VMware documentation website.

Create an anti-affinity rule by running the following command:

Example command

```
$ govc cluster.rule.create \
-name openshift4-control-plane-group \
-dc MyDatacenter -cluster MyCluster \
-enable \
-anti-affinity master-0 master-1 master-2
```

After creating the rule, your control plane nodes are automatically migrated by vSphere so they are not running on the same hosts. This might take some time while vSphere reconciles the new rule. Successful command completion is shown in the following procedure.



NOTE

The migration occurs automatically and might cause brief OpenShift API outage or latency until the migration finishes.

The vSphere DRS anti-affinity rules need to be updated manually in the event of a control plane VM name change or migration to a new vSphere Cluster.

Procedure

1. Remove any existing DRS anti-affinity rule by running the following command:

```
$ govc cluster.rule.remove \
-name openshift4-control-plane-group \
-dc MyDatacenter -cluster MyCluster
```

Example Output

```
[13-10-22 09:33:24] Reconfigure /MyDatacenter/host/MyCluster...OK
```

2. Create the rule again with updated names by running the following command:

```
$ govc cluster.rule.create \
-name openshift4-control-plane-group \
-dc MyDatacenter -cluster MyOtherCluster \
-enable \
-anti-affinity master-0 master-1 master-2
```

3.4.19. Telemetry access for OpenShift Container Platform

To provide metrics about cluster health and the success of updates, the Telemetry service requires internet access. When connected, this service runs automatically by default and registers your cluster to [OpenShift Cluster Manager](#).

After you confirm that your [OpenShift Cluster Manager](#) inventory is correct, either maintained automatically by Telemetry or manually by using OpenShift Cluster Manager, use subscription watch to track your OpenShift Container Platform subscriptions at the account or multi-cluster level. For more information about subscription watch, see "Data Gathered and Used by Red Hat's subscription services" in the *Additional resources* section.

Additional resources

- See [About remote health monitoring](#) for more information about the Telemetry service

3.4.20. Next steps

- [Customize your cluster](#).
- If necessary, you can [Remote health reporting](#).
- [Set up your registry and configure registry storage](#).
- Optional: [View the events from the vSphere Problem Detector Operator](#) to determine if the cluster has permission or storage configuration issues.
- Optional: if you created encrypted virtual machines, [create an encrypted storage class](#).

3.5. INSTALLING A CLUSTER ON VSphere IN A DISCONNECTED ENVIRONMENT WITH USER-PROVISIONED INFRASTRUCTURE

In OpenShift Container Platform version 4.19, you can install a cluster on VMware vSphere infrastructure that you provision in a restricted network.



IMPORTANT

The steps for performing a user-provisioned infrastructure installation are provided as an example only. Installing a cluster with infrastructure you provide requires knowledge of the vSphere platform and the installation process of OpenShift Container Platform. Use the user-provisioned infrastructure installation instructions as a guide; you are free to create the required resources through other methods.

3.5.1. Prerequisites

- You have completed the tasks in [Preparing to install a cluster using user-provisioned infrastructure](#).
- You reviewed your VMware platform licenses. Red Hat does not place any restrictions on your VMware licenses, but some VMware infrastructure components require licensing.
- You reviewed details about the [OpenShift Container Platform installation and update](#) processes.
- You read the documentation on [selecting a cluster installation method and preparing it for users](#).
- You [created a registry on your mirror host](#) and obtained the **imageContentSources** data for your version of OpenShift Container Platform.



IMPORTANT

Because the installation media is on the mirror host, you can use that computer to complete all installation steps.

- You provisioned [persistent storage](#) for your cluster. To deploy a private image registry, your storage must provide **ReadWriteMany** access modes.
- Completing the installation requires that you upload the Red Hat Enterprise Linux CoreOS (RHCOS) OVA on vSphere hosts. The machine from which you complete this process requires access to port 443 on the vCenter and ESXi hosts. You verified that port 443 is accessible.
- If you use a firewall, you confirmed with the administrator that port 443 is accessible. Control plane nodes must be able to reach vCenter and ESXi hosts on port 443 for the installation to succeed.
- If you use a firewall and plan to use the Telemetry service, you [configured the firewall to allow the sites](#) that your cluster requires access to.



NOTE

Be sure to also review this site list if you are configuring a proxy.

3.5.2. About installations in restricted networks

In OpenShift Container Platform 4.19, you can perform an installation that does not require an active connection to the internet to obtain software components. Restricted network installations can be completed using installer-provisioned infrastructure or user-provisioned infrastructure, depending on the cloud platform to which you are installing the cluster.

If you choose to perform a restricted network installation on a cloud platform, you still require access to its cloud APIs. Some cloud functions, like Amazon Web Service’s Route 53 DNS and IAM services, require internet access. Depending on your network, you might require less internet access for an installation on bare metal hardware, Nutanix, or on VMware vSphere.

To complete a restricted network installation, you must create a registry that mirrors the contents of the OpenShift image registry and contains the installation media. You can create this registry on a mirror host, which can access both the internet and your closed network, or by using other methods that meet your restrictions.



IMPORTANT

Because of the complexity of the configuration for user-provisioned installations, consider completing a standard user-provisioned infrastructure installation before you attempt a restricted network installation using user-provisioned infrastructure. Completing this test installation might make it easier to isolate and troubleshoot any issues that might arise during your installation in a restricted network.

3.5.2.1. Additional limits

Clusters in restricted networks have the following additional limitations and restrictions:

- The **ClusterVersion** status includes an **Unable to retrieve available updates** error.
- By default, you cannot use the contents of the Developer Catalog because you cannot access the required image stream tags.

3.5.3. Internet access for OpenShift Container Platform

In OpenShift Container Platform 4.19, you require access to the internet to obtain the images that are necessary to install your cluster.

You must have internet access to perform the following actions:

- Access [OpenShift Cluster Manager](#) to download the installation program and perform subscription management. If the cluster has internet access and you do not disable Telemetry, that service automatically entitles your cluster.
- Access [Quay.io](#) to obtain the packages that are required to install your cluster.
- Obtain the packages that are required to perform cluster updates.

3.5.4. VMware vSphere region and zone enablement

You can deploy an OpenShift Container Platform cluster to multiple vSphere data centers. Each data center can run multiple clusters. This configuration reduces the risk of a hardware failure or network outage that can cause your cluster to fail. To enable regions and zones, you must define multiple failure domains for your OpenShift Container Platform cluster.



IMPORTANT

The VMware vSphere region and zone enablement feature requires the vSphere Container Storage Interface (CSI) driver as the default storage driver in the cluster. As a result, the feature is only available on a newly installed cluster.

For a cluster that was upgraded from a previous release, you must enable CSI automatic migration for the cluster. You can then configure multiple regions and zones for the upgraded cluster.

The default installation configuration deploys a cluster to a single vSphere data center. If you want to deploy a cluster to multiple vSphere data centers, you must create an installation configuration file that enables the region and zone feature.

The default **install-config.yaml** file includes **vcenters** and **failureDomains** fields, where you can specify multiple vSphere data centers and clusters for your OpenShift Container Platform cluster. You can leave these fields blank if you want to install an OpenShift Container Platform cluster in a vSphere environment that consists of single data center.

The following list describes terms associated with defining zones and regions for your cluster:

- Failure domain: Establishes the relationships between a region and zone. You define a failure domain by using vCenter objects, such as a **datastore** object. A failure domain defines the vCenter location for OpenShift Container Platform cluster nodes.
- Region: Specifies a vCenter data center. You define a region by using a tag from the **openshift-region** tag category.
- Zone: Specifies a vCenter cluster. You define a zone by using a tag from the **openshift-zone** tag category.



NOTE

If you plan on specifying more than one failure domain in your **install-config.yaml** file, you must create tag categories, zone tags, and region tags in advance of creating the configuration file.

You must create a vCenter tag for each vCenter data center, which represents a region. Additionally, you must create a vCenter tag for each cluster than runs in a data center, which represents a zone. After you create the tags, you must attach each tag to their respective data centers and clusters.

The following table outlines an example of the relationship among regions, zones, and tags for a configuration with multiple vSphere data centers running in a single VMware vCenter.

Data center (region)	Cluster (zone)	Tags
us-east	us-east-1	us-east-1a
		us-east-1b
	us-east-2	us-east-2a

Data center (region)	Cluster (zone)	Tags
		us-east-2b
us-west	us-west-1	us-west-1a
		us-west-1b
	us-west-2	us-west-2a
		us-west-2b

Additional resources

- [Additional VMware vSphere configuration parameters](#)
- [Deprecated VMware vSphere configuration parameters](#)
- [vSphere automatic migration](#)
- [VMware vSphere CSI Driver Operator](#)

3.5.5. Manually creating the installation configuration file

To customise your OpenShift Container Platform deployment and meet specific network requirements, manually create the installation configuration file. This ensures that the installation program uses your tailored settings rather than default values during the setup process.



IMPORTANT

The Cloud Controller Manager Operator performs a connectivity check on a provided hostname or IP address. Ensure that you specify a hostname or an IP address to a reachable vCenter server. If you provide metadata to a non-existent vCenter server, installation of the cluster fails at the bootstrap stage.

Prerequisites

- You have an SSH public key on your local machine for use with the installation program. You can use the key for SSH authentication onto your cluster nodes for debugging and disaster recovery.
- You have obtained the OpenShift Container Platform installation program and the pull secret for your cluster.
- Obtain the **imageContentSources** section from the output of the command to mirror the repository.
- Obtain the contents of the certificate for your mirror registry.

Procedure

1. Create an installation directory to store your required installation assets in:

```
$ mkdir <installation_directory>
```



IMPORTANT

You must create a directory. Some installation assets, such as 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.

2. Customize the provided sample **install-config.yaml** file template and save the file in the **<installation_directory>**.



NOTE

You must name this configuration file **install-config.yaml**.

- Unless you use a registry that RHCOS trusts by default, such as **docker.io**, you must provide the contents of the certificate for your mirror repository in the **additionalTrustBundle** section. In most cases, you must provide the certificate for your mirror.
- You must include the **imageContentSources** section from the output of the command to mirror the repository.



IMPORTANT

- The **ImageContentSourcePolicy** file is generated as an output of **oc mirror** after the mirroring process is finished.
- The **oc mirror** command generates an **ImageContentSourcePolicy** file which contains the YAML needed to define **ImageContentSourcePolicy**. Copy the text from this file and paste it into your **install-config.yaml** file.
- You must run the 'oc mirror' command twice. The first time you run the **oc mirror** command, you get a full **ImageContentSourcePolicy** file. The second time you run the **oc mirror** command, you only get the difference between the first and second run. Because of this behavior, you must always keep a backup of these files in case you need to merge them into one complete **ImageContentSourcePolicy** file. Keeping a backup of these two output files ensures that you have a complete **ImageContentSourcePolicy** file.

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



IMPORTANT

Back up the **install-config.yaml** file now, because the installation process consumes the file in the next step.

Additional resources

- [Installation configuration parameters](#)

3.5.5.1. Sample `install-config.yaml` file for VMware vSphere

You can customize the **install-config.yaml** file to specify more details about your OpenShift Container Platform cluster's platform or modify the values of the required parameters.

```
additionalTrustBundlePolicy: Proxyonly
apiVersion: v1
baseDomain: example.com 1
compute: 2
- architecture: amd64
  name: <worker_node>
  platform: {}
  replicas: 0 3
controlPlane: 4
  architecture: amd64
  name: <parent_node>
  platform: {}
  replicas: 3 5
metadata:
  creationTimestamp: null
  name: test 6
networking:
---
platform:
  vsphere:
    failureDomains: 7
    - name: <failure_domain_name>
      region: <default_region_name>
      server: <fully_qualified_domain_name>
    topology:
      computeCluster: "/<data_center>/host/<cluster>" 8
      datacenter: <data_center> 9
      datastore: "/<data_center>/datastore/<datastore>" 10
      networks:
        - <VM_Network_name>
        resourcePool: "/<data_center>/host/<cluster>/Resources/<resourcePool>" 11
        folder: "/<data_center_name>/vm/<folder_name>/<subfolder_name>" 12
      zone: <default_zone_name>
    vcenters:
      - datacenters:
        - <data_center>
        password: <password> 13
        port: 443
        server: <fully_qualified_domain_name> 14
        user: administrator@vsphere.local
        diskType: thin 15
      tips: false 16
      pullSecret: '{"auths": {"<local_registry>": {"auth": "<credentials>","email": "you@example.com"}}}' 17
      sshKey: 'ssh-ed25519 AAAA...' 18
```

```
additionalTrustBundle: | 18
-----BEGIN CERTIFICATE-----
ZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZ
-----END CERTIFICATE-----
imageContentSources: 19
- mirrors:
  - <mirror_host_name>:<mirror_port>/<repo_name>/release
    source: <source_image_1>
- mirrors:
  - <mirror_host_name>:<mirror_port>/<repo_name>/release-images
    source: <source_image_2>
```

- 1 The base domain of the cluster. All DNS records must be sub-domains of this base and include the cluster name.
- 2 4 The **controlPlane** section is a single mapping, but the compute section is a sequence of mappings. To meet the requirements of the different data structures, the first line of the **compute** section must begin with a hyphen, -, and the first line of the **controlPlane** section must not. Both sections define a single machine pool, so only one control plane is used. OpenShift Container Platform does not support defining multiple compute pools.
- 3 You must set the value of the **replicas** parameter to **0**. This parameter controls the number of workers that the cluster creates and manages for you, which are functions that the cluster does not perform when you use user-provisioned infrastructure. You must manually deploy worker machines for the cluster to use before you finish installing OpenShift Container Platform.
- 5 The number of control plane machines that you add to the cluster. Because the cluster uses this values as the number of etcd endpoints in the cluster, the value must match the number of control plane machines that you deploy.
- 6 The cluster name that you specified in your DNS records.
- 7 Establishes the relationships between a region and zone. You define a failure domain by using vCenter objects, such as a **datastore** object. A failure domain defines the vCenter location for OpenShift Container Platform cluster nodes.
- 8 The vSphere data center.
- 9 The path to the vSphere datastore that holds virtual machine files, templates, and ISO images.

IMPORTANT

You can specify the path of any datastore that exists in a datastore cluster. By default, Storage vMotion is automatically enabled for a datastore cluster. Red Hat does not support Storage vMotion, so you must disable Storage vMotion to avoid data loss issues for your OpenShift Container Platform cluster.

If you must specify VMs across multiple datastores, use a **datastore** object to specify a failure domain in your cluster's **install-config.yaml** configuration file. For more information, see "VMware vSphere region and zone enablement".

10 Optional: For installer-provisioned infrastructure, the absolute path of an existing resource pool where the installation program creates the virtual machines, for example, `/<data_center_name>/host/<cluster_name>/Resources/<resource_pool_name>/<optional_nested_resource_pool_name>`. If you do not specify a value, resources are installed in the root of

the cluster **/example_data_center/host/example_cluster/Resources**.

- 11 Optional: For installer-provisioned infrastructure, the absolute path of an existing folder where the installation program creates the virtual machines, for example, `/<data_center_name>/vm/<folder_name>/<subfolder_name>`. If you do not provide this value, the installation program creates a top-level folder in the data center virtual machine folder that is named with the infrastructure ID. If you are providing the infrastructure for the cluster and you do not want to use the default **StorageClass** object, named **thin**, you can omit the **folder** parameter from the **install-config.yaml** file.
- 12 The password associated with the vSphere user.
- 13 The fully-qualified hostname or IP address of the vCenter server.



IMPORTANT

The Cloud Controller Manager Operator performs a connectivity check on a provided hostname or IP address. Ensure that you specify a hostname or an IP address to a reachable vCenter server. If you provide metadata to a non-existent vCenter server, installation of the cluster fails at the bootstrap stage.

- 14 The vSphere disk provisioning method.
- 15 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.



IMPORTANT

To enable FIPS mode for your cluster, you must run the installation program from a Red Hat Enterprise Linux (RHEL) computer configured to operate in FIPS mode. For more information about configuring FIPS mode on RHEL, see [Switching RHEL to FIPS mode](#).

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.

- 16 For **<local_registry>**, specify the registry domain name, and optionally the port, that your mirror registry uses to serve content. For example **registry.example.com** or **registry.example.com:5000**. For **<credentials>**, specify the base64-encoded user name and password for your mirror registry.
- 17 The public portion of the default SSH key for the **core** user in Red Hat Enterprise Linux CoreOS (RHCOS).



NOTE

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

- 18 Provide the contents of the certificate file that you used for your mirror registry.
- 19 Provide the **imageContentSources** section from the output of the command to mirror the repository.

3.5.5.2. Configuring the cluster-wide proxy during installation

To enable internet access in environments that deny direct connections, configure a cluster-wide proxy in the **install-config.yaml** file. This configuration ensures that the new OpenShift Container Platform cluster routes traffic through the specified HTTP or HTTPS proxy.

Prerequisites

- You have an existing **install-config.yaml** file.
- You have reviewed the sites that your cluster requires access to and determined whether any of them need to bypass the proxy. By default, all cluster egress traffic is proxied, including calls to hosting cloud provider APIs. You added sites to the **Proxy** object's **spec.noProxy** field to bypass the proxy if necessary.



NOTE

The **Proxy** object **status.noProxy** field is populated with the values of the **networking.machineNetwork[].cidr**, **networking.clusterNetwork[].cidr**, and **networking.serviceNetwork[]** fields from your installation configuration.

For installations on Amazon Web Services (AWS), Google Cloud, Microsoft Azure, and Red Hat OpenStack Platform (RHOSP), the **Proxy** object **status.noProxy** field is also populated with the instance metadata endpoint (169.254.169.254).

Procedure

1. Edit your **install-config.yaml** file and add the proxy settings. For example:

```
apiVersion: v1
baseDomain: my.domain.com
proxy:
  httpProxy: http://<username>:<pswd>@<ip>:<port>
  httpsProxy: https://<username>:<pswd>@<ip>:<port>
  noProxy: example.com
  additionalTrustBundle: |
    -----BEGIN CERTIFICATE-----
    <MY_TRUSTED_CA_CERT>
    -----END CERTIFICATE-----
  additionalTrustBundlePolicy: <policy_to_add_additionalTrustBundle>
  # ...
```

where:

proxy.httpProxy

Specifies a proxy URL to use for creating HTTP connections outside the cluster. The URL scheme must be **http**.

proxy.httpsProxy

Specifies a proxy URL to use for creating HTTPS connections outside the cluster.

proxy.noProxy

Specifies a comma-separated list of destination domain names, IP addresses, or other network CIDRs to exclude from proxying. Preface a domain with `.` to match subdomains only. For example, `.y.com` matches `x.y.com`, but not `y.com`. Use `*` to bypass the proxy for all destinations. You must include vCenter's IP address and the IP range that you use for its machines.

additionalTrustBundle

If provided, the installation program generates a config map that is named `user-ca-bundle` in the `openshift-config` namespace to hold the additional CA certificates. If you provide `additionalTrustBundle` and at least one proxy setting, the `Proxy` object is configured to reference the `user-ca-bundle` config map in the `trustedCA` field. The Cluster Network Operator then creates a `trusted-ca-bundle` config map that merges the contents specified for the `trustedCA` parameter with the RHCOS trust bundle. The `additionalTrustBundle` field is required unless the proxy's identity certificate is signed by an authority from the RHCOS trust bundle.

additionalTrustBundlePolicy

Specifies the policy that determines the configuration of the `Proxy` object to reference the `user-ca-bundle` config map in the `trustedCA` field. The allowed values are `Proxyonly` and `Always`. Use `Proxyonly` to reference the `user-ca-bundle` config map only when `http/https` proxy is configured. Use `Always` to always reference the `user-ca-bundle` config map. The default value is `Proxyonly`. Optional parameter.



NOTE

The installation program does not support the proxy `readinessEndpoints` field.



NOTE

If the installer times out, restart and then complete the deployment by using the `wait-for` command of the installer. For example:

+

```
$ ./openshift-install wait-for install-complete --log-level debug
```

2. Save the file and reference it when installing OpenShift Container Platform.

The installation program creates a cluster-wide proxy that is named `cluster` that uses the proxy settings in the provided `install-config.yaml` file. If no proxy settings are provided, a `cluster` `Proxy` object is still created, but it will have a nil `spec`.



NOTE

Only the `Proxy` object named `cluster` is supported, and no additional proxies can be created.

3.5.5.3. Configuring regions and zones for a VMware vCenter

You can modify the default installation configuration file, so that you can deploy an OpenShift Container Platform cluster to multiple vSphere data centers.

The default **install-config.yaml** file configuration from the previous release of OpenShift Container Platform is deprecated. You can continue to use the deprecated default configuration, but the **openshift-installer** will prompt you with a warning message that indicates the use of deprecated fields in the configuration file.

Prerequisites

- You have an existing **install-config.yaml** installation configuration file.



IMPORTANT

You must specify at least one failure domain for your OpenShift Container Platform cluster, so that you can provision data center objects for your VMware vCenter server. Consider specifying multiple failure domains if you need to provision virtual machine nodes in different data centers, clusters, datastores, and other components. To enable regions and zones, you must define multiple failure domains for your OpenShift Container Platform cluster.

- You have installed the **govc** command line tool.



IMPORTANT

The example uses the **govc** command. The **govc** command is an open source command available from VMware; it is not available from Red Hat. The Red Hat support team does not maintain the **govc** command. Instructions for downloading and installing **govc** are found on the VMware documentation website.

Procedure

1. Create the **openshift-region** and **openshift-zone** vCenter tag categories by running the following commands:



IMPORTANT

If you specify different names for the **openshift-region** and **openshift-zone** vCenter tag categories, the installation of the OpenShift Container Platform cluster fails.

```
$ govc tags.category.create -d "OpenShift region" openshift-region
```

```
$ govc tags.category.create -d "OpenShift zone" openshift-zone
```

2. For each region where you want to deploy your cluster, create a region tag by running the following command:

```
$ govc tags.create -c <region_tag_category> <region_tag>
```

3. For each zone where you want to deploy your cluster, create a zone tag by running the following command:

```
$ govc tags.create -c <zone_tag_category> <zone_tag>
```

4. Attach region tags to each vCenter data center object by running the following command:

```
$ govc tags.attach -c <region_tag_category> <region_tag_1> /<data_center_1>
```

5. Attach the zone tags to each vCenter cluster object by running the following command:

```
$ govc tags.attach -c <zone_tag_category> <zone_tag_1> /<data_center_1>/host/<cluster1>
```

6. Change to the directory that contains the installation program and initialize the cluster deployment according to your chosen installation requirements.

Sample `install-config.yaml` file with multiple data centers defined in a vSphere center

```
# ...
compute:
---
vsphere:
  zones:
    - "<machine_pool_zone_1>"
    - "<machine_pool_zone_2>"
# ...
controlPlane:
# ...
vsphere:
  zones:
    - "<machine_pool_zone_1>"
    - "<machine_pool_zone_2>"
# ...
platform:
  vsphere:
    vcenters:
# ...
  datacenters:
    - <data_center_1_name>
    - <data_center_2_name>
failureDomains:
  - name: <machine_pool_zone_1>
    region: <region_tag_1>
    zone: <zone_tag_1>
    server: <fullyQualifiedDomainName>
topology:
  datacenter: <data_center_1>
  computeCluster: "/<data_center_1>/host/<cluster1>"
  networks:
    - <VM_Network1_name>
    datastore: "/<data_center_1>/datastore/<datastore1>"
    resourcePool: "/<data_center_1>/host/<cluster1>/Resources/<resourcePool1>"
    folder: "/<data_center_1>/vm/<folder1>"
  - name: <machine_pool_zone_2>
    region: <region_tag_2>
```

```

zone: <zone_tag_2>
server: <fully_qualified_domain_name>
topology:
  datacenter: <data_center_2>
  computeCluster: "/<data_center_2>/host/<cluster2>" 
  networks:
    - <VM_Network2_name>
  datastore: "/<data_center_2>/datastore/<datastore2>" 
  resourcePool: "/<data_center_2>/host/<cluster2>/Resources/<resourcePool2>" 
  folder: "/<data_center_2>/vm/<folder2>" 
# ...

```

3.5.6. Creating the Kubernetes manifest and Ignition config files

To customize cluster definitions and manually start machines, generate the Kubernetes manifest and Ignition config files. These assets provide the necessary instructions to configure the cluster infrastructure according to your specific deployment requirements.

The installation configuration file transforms into the Kubernetes manifests. The manifests wrap into the Ignition configuration files, which are later used to configure the cluster machines.



IMPORTANT

- The Ignition config files that the OpenShift Container Platform 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.
- It is recommended that you use 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.

Prerequisites

- You obtained the OpenShift Container Platform installation program. For a restricted network installation, these files are on your mirror host.
- You created the **install-config.yaml** installation configuration file.

Procedure

1. Change to the directory that contains the OpenShift Container Platform installation program and generate the Kubernetes manifests for the cluster:

```
$ ./openshift-install create manifests --dir <installation_directory>
```

where

<installation_directory>

Specifies the installation directory that contains the **install-config.yaml** file you created.

2. Remove the Kubernetes manifest files that define the control plane machines, compute machine sets, and control plane machine sets:

```
$ rm -f openshift/99_openshift-cluster-api_master-machines-*.yaml openshift/99_openshift-cluster-api_worker-machineset-*.yaml openshift/99_openshift-machine-api_master-control-plane-machine-set.yaml
```

Because you create and manage these resources yourself, you do not have to initialize them. You can preserve the compute machine set files to create compute machines by using the machine API, but you must update references to them to match your environment.

3. Check that the **mastersSchedulable** parameter in the **<installation_directory>/manifests/cluster-scheduler-02-config.yaml** Kubernetes manifest file is set to **false**. This setting prevents pods from being scheduled on the control plane machines:
 - a. Open the **<installation_directory>/manifests/cluster-scheduler-02-config.yaml** file.
 - b. Locate the **mastersSchedulable** parameter and ensure that it is set to **false**.
 - c. Save and exit the file.
4. To create the Ignition configuration files, run the following command from the directory that contains the installation program:

```
$ ./openshift-install create ignition-configs --dir <installation_directory>
```

where:

<installation_directory>

Specifies the same installation directory.

Ignition config files are created for the bootstrap, control plane, and compute nodes in the installation directory. The **kubeadmin-password** and **kubeconfig** files are created in the **./<installation_directory>/auth** directory:

```

    auth
    |
    +-- kubeadmin-password
    |
    +-- kubeconfig
    |
    +-- bootstrap.ign
    |
    +-- master.ign
    |
    +-- metadata.json
    |
    +-- worker.ign

```

3.5.7. Configuring chrony time service

You must set the time server and related settings used by the chrony time service (**chrony**) by modifying the contents of the **chrony.conf** file and passing those contents to your nodes as a machine config.

Procedure

1. Create a Butane config including the contents of the **chrony.conf** file. For example, to configure chrony on worker nodes, create a **99-worker-chrony.bu** file.



NOTE

The [Butane version](#) you specify in the config file should match the OpenShift Container Platform version and always ends in **0**. For example, **4.19.0**. See "Creating machine configs with Butane" for information about Butane.

```
variant: openshift
version: 4.19.0
metadata:
  name: 99-worker-chrony 1
  labels:
    machineconfiguration.openshift.io/role: worker 2
storage:
  files:
    - path: /etc/chrony.conf
      mode: 0644 3
      overwrite: true
  contents:
    inline: |
      pool 0.rhel.pool.ntp.org iburst 4
      driftfile /var/lib/chrony/drift
      makestep 1.0 3
      rtcsync
      logdir /var/log/chrony
```

1 **2** On control plane nodes, substitute **master** for **worker** in both of these locations.

3 Specify an octal value mode for the **mode** field in the machine config file. After creating the file and applying the changes, the **mode** is converted to a decimal value. You can check the YAML file with the command **oc get mc <mc-name> -o yaml**.

4 Specify any valid, reachable time source, such as the one provided by your DHCP server.



NOTE

For all-machine to all-machine communication, the Network Time Protocol (NTP) on UDP is port **123**. If an external NTP time server is configured, you must open UDP port **123**.

2. Use Butane to generate a **MachineConfig** object file, **99-worker-chrony.yaml**, containing the configuration to be delivered to the nodes:

```
$ butane 99-worker-chrony.bu -o 99-worker-chrony.yaml
```

3. Apply the configurations in one of two ways:

- If the cluster is not running yet, after you generate manifest files, add the **MachineConfig** object file to the **<installation_directory>/openshift** directory, and then continue to create the cluster.

- If the cluster is already running, apply the file:

```
$ oc apply -f ./99-worker-chrony.yaml
```

3.5.8. Extracting the infrastructure name

The Ignition config files contain a unique cluster identifier that you can use to uniquely identify your cluster in VMware vSphere. If you plan to use the cluster identifier as the name of your virtual machine folder, you must extract it.

Prerequisites

- You obtained the OpenShift Container Platform installation program and the pull secret for your cluster.
- You generated the Ignition config files for your cluster.
- You installed the **jq** package.

Procedure

- To extract and view the infrastructure name from the Ignition config file metadata, run the following command:

```
$ jq -r .infracID <installation_directory>/metadata.json ①
```

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

Example output

```
openshift-vw9j6 ①
```

① The output of this command is your cluster name and a random string.

3.5.9. Installing RHCOS and starting the OpenShift Container Platform bootstrap process

To install OpenShift Container Platform on user-provisioned infrastructure on VMware vSphere, you must install Red Hat Enterprise Linux CoreOS (RHCOS) on vSphere hosts. When you install RHCOS, you must provide the Ignition config file that was generated by the OpenShift Container Platform installation program for the type of machine you are installing. If you have configured suitable networking, DNS, and load balancing infrastructure, the OpenShift Container Platform bootstrap process begins automatically after the RHCOS machines have rebooted.

Prerequisites

- You have obtained the Ignition config files for your cluster.
- You have access to an HTTP server that you can access from your computer and that the machines that you create can access.

- You have created a [vSphere cluster](#).

Procedure

1. Upload the bootstrap Ignition config file, which is named **<installation_directory>/bootstrap.ign**, that the installation program created to your HTTP server. Note the URL of this file.
2. Save the following secondary Ignition config file for your bootstrap node to your computer as **<installation_directory>/merge-bootstrap.ign**:

```
{
  "ignition": {
    "config": {
      "merge": [
        {
          "source": "<bootstrap_ignition_config_url>", ①
          "verification": {}
        }
      ],
      "timeouts": {},
      "version": "3.2.0"
    },
    "networkd": {},
    "passwd": {},
    "storage": {},
    "systemd": {}
  }
}
```

- 1 Specify the URL of the bootstrap Ignition config file that you hosted.

When you create the virtual machine (VM) for the bootstrap machine, you use this Ignition config file.

3. Locate the following Ignition config files that the installation program created:
 - **<installation_directory>/master.ign**
 - **<installation_directory>/worker.ign**
 - **<installation_directory>/merge-bootstrap.ign**
4. Convert the Ignition config files to Base64 encoding. Later in this procedure, you must add these files to the extra configuration parameter **guestinfo.ignition.config.data** in your VM. For example, if you use a Linux operating system, you can use the **base64** command to encode the files.

```
$ base64 -w0 <installation_directory>/master.ign > <installation_directory>/master.64
```

```
$ base64 -w0 <installation_directory>/worker.ign > <installation_directory>/worker.64
```

```
$ base64 -w0 <installation_directory>/merge-bootstrap.ign > <installation_directory>/merge-bootstrap.64
```



IMPORTANT

If you plan to add more compute machines to your cluster after you finish installation, do not delete these files.

5. Obtain the RHCOS OVA image. Images are available from the [RHCOS image mirror](#) page.



IMPORTANT

The RHCOS images might not change with every release of OpenShift Container Platform. You must download an image with the highest version that is less than or equal to the OpenShift Container Platform version that you install. Use the image version that matches your OpenShift Container Platform version if it is available.

The filename contains the OpenShift Container Platform version number in the format **rhcos-vmware.<architecture>.ova**.

6. In the vSphere Client, create a folder in your data center to store your VMs.
 - a. Click the **VMs and Templates** view.
 - b. Right-click the name of your data center.
 - c. Click **New Folder** → **New VM and Template Folder**.
7. In the vSphere Client, create a template for the OVA image and then clone the template as needed.



NOTE

In the following steps, you create a template and then clone the template for all of your cluster machines. You then provide the location for the Ignition config file for that cloned machine type when you provision the VMs.

- a. From the **Hosts and Clusters** tab, right-click your cluster name and select **Deploy OVF Template**.
- b. On the **Select an OVF** tab, specify the name of the RHCOS OVA file that you downloaded.
- c. On the **Select a name and folder** tab, set a **Virtual machine name** for your template, such as **Template-RHCOS**. Click the name of your vSphere cluster and select the folder you created in the previous step.
- d. On the **Select a compute resource** tab, click the name of your vSphere cluster.
- e. On the **Select storage** tab, configure the storage options for your VM.

- Select **Thin Provision** or **Thick Provision**, based on your storage preferences.
- Select the datastore that you specified in your **install-config.yaml** file.
- If you want to encrypt your virtual machines, select **Encrypt this virtual machine**. See the section titled "Requirements for encrypting virtual machines" for more information.

f. On the **Select network** tab, specify the network that you configured for the cluster, if available.

g. When creating the OVF template, do not specify values on the **Customize template** tab or configure the template any further.



IMPORTANT

Do not start the original VM template. The VM template must remain off and must be cloned for new RHCOS machines. Starting the VM template configures the VM template as a VM on the platform, which prevents it from being used as a template that compute machine sets can apply configurations to.

8. Optional: Update the configured virtual hardware version in the VM template, if necessary. Follow [Upgrading a virtual machine to the latest hardware version](#) in the VMware documentation for more information.



IMPORTANT

It is recommended that you update the hardware version of the VM template to version 15 before creating VMs from it, if necessary. Using hardware version 13 for your cluster nodes running on vSphere is now deprecated. If your imported template defaults to hardware version 13, you must ensure that your ESXi host is on 6.7U3 or later before upgrading the VM template to hardware version 15. If your vSphere version is less than 6.7U3, you can skip this upgrade step; however, a future version of OpenShift Container Platform is scheduled to remove support for hardware version 13 and vSphere versions less than 6.7U3.

9. After the template deploys, deploy a VM for a machine in the cluster.

- Right-click the template name and click **Clone** → **Clone to Virtual Machine**
- On the **Select a name and folder** tab, specify a name for the VM. You might include the machine type in the name, such as **control-plane-0** or **compute-1**.



NOTE

Ensure that all virtual machine names across a vSphere installation are unique.

- On the **Select a name and folder** tab, select the name of the folder that you created for the cluster.
- On the **Select a compute resource** tab, select the name of a host in your data center.
- On the **Select clone options** tab, select **Customize this virtual machine's hardware**

f. On the **Customize hardware** tab, click **Advanced Parameters**.



IMPORTANT

The following configuration suggestions are for example purposes only. As a cluster administrator, you must configure resources according to the resource demands placed on your cluster. To best manage cluster resources, consider creating a resource pool from the cluster's root resource pool.

- Optional: Override default DHCP networking in vSphere. To enable static IP networking:
 - Set your static IP configuration:

Example command

```
$ export IPCFG="ip=<ip>::<gateway>:<netmask>:<hostname>:<iface>:none
nameserver=srv1 [nameserver=srv2 [nameserver=srv3 [...]]]"
```

Example command

```
$ export IPCFG="ip=192.168.100.101::192.168.100.254:255.255.255.0::none
nameserver=8.8.8.8"
```

- Set the **guestinfo.afterburn.initrd.network-kargs** property before you boot a VM from an OVA in vSphere:

Example command

```
$ govc vm.change -vm "<vm_name>" -e "guestinfo.afterburn.initrd.network-
kargs=${IPCFG}"
```

- Add the following configuration parameter names and values by specifying data in the **Attribute** and **Values** fields. Ensure that you select the **Add** button for each parameter that you create.
 - **guestinfo.ignition.config.data**: Locate the base-64 encoded files that you created previously in this procedure, and paste the contents of the base64-encoded Ignition config file for this machine type.
 - **guestinfo.ignition.config.data.encoding**: Specify **base64**.
 - **disk.EnableUUID**: Specify **TRUE**.
 - **stealclock.enable**: If this parameter was not defined, add it and specify **TRUE**.
 - Create a child resource pool from the cluster's root resource pool. Perform resource allocation in this child resource pool.

g. In the **Virtual Hardware** panel of the **Customize hardware** tab, modify the specified values as required. Ensure that the amount of RAM, CPU, and disk storage meets the minimum requirements for the machine type.

- h. Complete the remaining configuration steps. On clicking the **Finish** button, you have completed the cloning operation.
 - i. From the **Virtual Machines** tab, right-click on your VM and then select **Power → Power On**.
 - j. Check the console output to verify that Ignition ran.

Example command

```
Ignition: ran on 2022/03/14 14:48:33 UTC (this boot)
Ignition: user-provided config was applied
```

Next steps

- Create the rest of the machines for your cluster by following the preceding steps for each machine.



IMPORTANT

You must create the bootstrap and control plane machines at this time. Because some pods are deployed on compute machines by default, also create at least two compute machines before you install the cluster.

3.5.10. Adding more compute machines to a cluster in vSphere

You can add more compute machines to a user-provisioned OpenShift Container Platform cluster on VMware vSphere.

After your vSphere template deploys in your OpenShift Container Platform cluster, you can deploy a virtual machine (VM) for a machine in that cluster.

Prerequisites

- Obtain the base64-encoded Ignition file for your compute machines.
- You have access to the vSphere template that you created for your cluster.

Procedure

1. Right-click the template's name and click **Clone → Clone to Virtual Machine**
2. On the **Select a name and folder** tab, specify a name for the VM. You might include the machine type in the name, such as **compute-1**.



NOTE

Ensure that all virtual machine names across a vSphere installation are unique.

3. On the **Select a name and folder** tab, select the name of the folder that you created for the cluster.
4. On the **Select a compute resource** tab, select the name of a host in your data center.
5. On the **Select storage** tab, select storage for your configuration and disk files.

6. On the **Select clone options** tab, select **Customize this virtual machine's hardware**.
7. On the **Customize hardware** tab, click **Advanced Parameters**.
 - Add the following configuration parameter names and values by specifying data in the **Attribute** and **Values** fields. Ensure that you select the **Add** button for each parameter that you create.
 - **guestinfo.ignition.config.data**: Paste the contents of the base64-encoded compute Ignition config file for this machine type.
 - **guestinfo.ignition.config.data.encoding**: Specify **base64**.
 - **disk.EnableUUID**: Specify **TRUE**.
8. In the **Virtual Hardware** panel of the **Customize hardware** tab, modify the specified values as required. Ensure that the amount of RAM, CPU, and disk storage meets the minimum requirements for the machine type. If many networks exist, select **Add New Device** > **Network Adapter**, and then enter your network information in the fields provided by the **New Network** menu item.
9. Complete the remaining configuration steps. On clicking the **Finish** button, you have completed the cloning operation.
10. From the **Virtual Machines** tab, right-click on your VM and then select **Power** → **Power On**.

Next steps

- Continue to create more compute machines for your cluster.

3.5.11. Disk partitioning

In most cases, data partitions are originally created by installing RHCOS, rather than by installing another operating system. In such cases, the OpenShift Container Platform installer should be allowed to configure your disk partitions.

However, there are two cases where you might want to intervene to override the default partitioning when installing an OpenShift Container Platform node:

- Create separate partitions: For greenfield installations on an empty disk, you might want to add separate storage to a partition. This is officially supported for making **/var** or a subdirectory of **/var**, such as **/var/lib/etcd**, a separate partition, but not both.



IMPORTANT

For disk sizes larger than 100GB, and especially disk sizes larger than 1TB, create a separate **/var** partition. See "Creating a separate **/var** partition" and this [Red Hat Knowledgebase article](#) for more information.



IMPORTANT

Kubernetes supports only two file system partitions. If you add more than one partition to the original configuration, Kubernetes cannot monitor all of them.

- Retain existing partitions: For a brownfield installation where you are reinstalling OpenShift

Container Platform on an existing node and want to retain data partitions installed from your previous operating system, there are both boot arguments and options to **coreos-installer** that allow you to retain existing data partitions.

3.5.12. Creating a separate `/var` partition

In general, disk partitioning for OpenShift Container Platform should be left to the installer. However, there are cases where you might want to create separate partitions in a part of the filesystem that you expect to grow.

OpenShift Container Platform supports the addition of a single partition to attach storage to either the `/var` partition or a subdirectory of `/var`. For example:

- **`/var/lib/containers`:** Holds container-related content that can grow as more images and containers are added to a system.
- **`/var/lib/etcd`:** Holds data that you might want to keep separate for purposes such as performance optimization of etcd storage.
- **`/var`:** Holds data that you might want to keep separate for purposes such as auditing.



IMPORTANT

For disk sizes larger than 100GB, and especially larger than 1TB, create a separate `/var` partition.

Storing the contents of a `/var` directory separately makes it easier to grow storage for those areas as needed and reinstall OpenShift Container Platform at a later date and keep that data intact. With this method, you will not have to pull all your containers again, nor will you have to copy massive log files when you update systems.

Because `/var` must be in place before a fresh installation of Red Hat Enterprise Linux CoreOS (RHCOS), the following procedure sets up the separate `/var` partition by creating a machine config manifest that is inserted during the **openshift-install** preparation phases of an OpenShift Container Platform installation.

Procedure

1. Create a directory to hold the OpenShift Container Platform installation files:

```
$ mkdir $HOME/clusterconfig
```

2. Run **openshift-install** to create a set of files in the `manifest` and `openshift` subdirectories. Answer the system questions as you are prompted:

```
$ openshift-install create manifests --dir $HOME/clusterconfig
? SSH Public Key ...
$ ls $HOME/clusterconfig/openshift/
99_kubeadmin-password-secret.yaml
99_openshift-cluster-api_master-machines-0.yaml
99_openshift-cluster-api_master-machines-1.yaml
99_openshift-cluster-api_master-machines-2.yaml
...
```

3. Create a Butane config that configures the additional partition. For example, name the file **\$HOME/clusterconfig/98-var-partition.bu**, change the disk device name to the name of the storage device on the **worker** systems, and set the storage size as appropriate. This example places the **/var** directory on a separate partition:

```

variant: openshift
version: 4.19.0
metadata:
  labels:
    machineconfiguration.openshift.io/role: worker
  name: 98-var-partition
storage:
  disks:
    - device: /dev/disk/by-id/<device_name> 1
      partitions:
        - label: var
          start_mib: <partition_start_offset> 2
          size_mib: <partition_size> 3
          number: 5
    filesystems:
      - device: /dev/disk/by-partlabel/var
        path: /var
        format: xfs
        mount_options: [defaults, prjquota] 4
        with_mount_unit: true

```

- 1 The storage device name of the disk that you want to partition.
- 2 When adding a data partition to the boot disk, a minimum value of 25000 mebibytes is recommended. The root file system is automatically resized to fill all available space up to the specified offset. If no value is specified, or if the specified value is smaller than the recommended minimum, the resulting root file system will be too small, and future reinstalls of RHCOS might overwrite the beginning of the data partition.
- 3 The size of the data partition in mebibytes.
- 4 The **prjquota** mount option must be enabled for filesystems used for container storage.



NOTE

When creating a separate **/var** partition, you cannot use different instance types for worker nodes, if the different instance types do not have the same device name.

4. Create a manifest from the Butane config and save it to the **clusterconfig/openshift** directory. For example, run the following command:

```
$ butane $HOME/clusterconfig/98-var-partition.bu -o $HOME/clusterconfig/openshift/98-var-partition.yaml
```

5. Run **openshift-install** again to create Ignition configs from a set of files in the **manifest** and **openshift** subdirectories:

```
$ openshift-install create ignition-configs --dir $HOME/clusterconfig
$ ls $HOME/clusterconfig/
auth bootstrap.ign master.ign metadata.json worker.ign
```

Now you can use the Ignition config files as input to the vSphere installation procedures to install Red Hat Enterprise Linux CoreOS (RHCOS) systems.

3.5.13. Waiting for the bootstrap process to complete

To install OpenShift Container Platform, use Ignition configuration files to initialize the bootstrap process after the cluster nodes boot into RHCOS. You must wait for this process to complete to ensure the cluster is fully installed.

Prerequisites

- You have created the Ignition config files for your cluster.
- You have configured suitable network, DNS, and load balancing infrastructure.
- You have obtained the installation program and generated the Ignition config files for your cluster.
- You installed RHCOS on your cluster machines and provided the Ignition config files that the OpenShift Container Platform installation program generated.

Procedure

1. Monitor the bootstrap process:

```
$ ./openshift-install --dir <installation_directory> wait-for bootstrap-complete \
--log-level=info
```

where:

<installation_directory>

Specifies the path to the directory that stores the installation files.

--log-level=info

Specifies **warn**, **debug**, or **error** instead of **info** to view different installation details.

Example output

```
INFO Waiting up to 30m0s for the Kubernetes API at https://api.test.example.com:6443...
INFO API v1.32.3 up
INFO Waiting up to 30m0s for bootstrapping to complete...
INFO It is now safe to remove the bootstrap resources
```

The command succeeds when the Kubernetes API server signals that it has been bootstrapped on the control plane machines.

2. After the bootstrap process is complete, remove the bootstrap machine from the load balancer.



IMPORTANT

You must remove the bootstrap machine from the load balancer at this point. You can also remove or reformat the bootstrap machine itself.

3.5.14. Logging in to the cluster by using the CLI

To log in to your cluster as the default system user, export the **kubeconfig** file. This configuration enables the CLI to authenticate and connect to the specific API server created during OpenShift Container Platform installation.

The **kubeconfig** file is specific to a cluster and is created during OpenShift Container Platform installation.

Prerequisites

- You deployed an OpenShift Container Platform cluster.
- You installed the OpenShift CLI (**oc**).

Procedure

1. Export the **kubeadmin** credentials by running the following command:

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

where:

<installation_directory>

Specifies the path to the directory that stores the installation files.

2. Verify you can run **oc** commands successfully using the exported configuration by running the following command:

```
$ oc whoami
```

Example output

```
system:admin
```

3.5.15. Approving the certificate signing requests for your machines

To add machines to a cluster, verify the status of the certificate signing requests (CSRs) generated for each machine. If manual approval is required, approve the client requests first, followed by the server requests.

Prerequisites

- You added machines to your cluster.

Procedure

1. Confirm that the cluster recognizes the machines:

```
$ oc get nodes
```

Example output

```
NAME STATUS ROLES AGE VERSION
master-0 Ready master 63m v1.32.3
master-1 Ready master 63m v1.32.3
master-2 Ready master 64m v1.32.3
```

The output lists all of the machines that you created.



NOTE

The preceding output might not include the compute nodes, also known as worker nodes, until some CSRs are approved.

2. Review the pending CSRs and ensure that you see the client requests with the **Pending** or **Approved** status for each machine that you added to the cluster:

```
$ oc get csr
```

Example output

```
NAME AGE REQUESTOR CONDITION
csr-8b2br 15m system:serviceaccount:openshift-machine-config-operator:node-
bootstrapper Pending
csr-8vnps 15m system:serviceaccount:openshift-machine-config-operator:node-
bootstrapper Pending
...
```

In this example, two machines are joining the cluster. You might see more approved CSRs in the list.

3. If the CSRs were not approved, after all of the pending CSRs for the machines you added are in **Pending** status, approve the CSRs for your cluster machines:



NOTE

Because the CSRs rotate automatically, approve your CSRs within an hour of adding the machines to the cluster. If you do not approve them within an hour, the certificates will rotate, and more than two certificates will be present for each node. You must approve all of these certificates. After the client CSR is approved, the Kubelet creates a secondary CSR for the serving certificate, which requires manual approval. Then, subsequent serving certificate renewal requests are automatically approved by the **machine-approver** if the Kubelet requests a new certificate with identical parameters.



NOTE

For clusters running on platforms that are not machine API enabled, such as bare metal and other user-provisioned infrastructure, you must implement a method of automatically approving the kubelet serving certificate requests (CSRs). If a request is not approved, then the **oc exec**, **oc rsh**, and **oc logs** commands cannot succeed, because a serving certificate is required when the API server connects to the kubelet. Any operation that contacts the Kubelet endpoint requires this certificate approval to be in place. The method must watch for new CSRs, confirm that the CSR was submitted by the **node-bootstrapper** service account in the **system:node** or **system:admin** groups, and confirm the identity of the node.

- To approve them individually, run the following command for each valid CSR:

```
$ oc adm certificate approve <csr_name>
```

where:

<csr_name>

Specifies the name of a CSR from the list of current CSRs.

- To approve all pending CSRs, run the following command:

```
$ oc get csr -o go-template='{{range .items}}{{if not .status}}{{.metadata.name}}\n{{end}}{{end}}' | xargs --no-run-if-empty oc adm certificate approve
```



NOTE

Some Operators might not become available until some CSRs are approved.

4. Now that your client requests are approved, you must review the server requests for each machine that you added to the cluster:

```
$ oc get csr
```

Example output

NAME	AGE	REQUESTOR	CONDITION
csr-bfd72	5m26s	system:node:ip-10-0-50-126.us-east-2.compute.internal	Pending
csr-c57lv	5m26s	system:node:ip-10-0-95-157.us-east-2.compute.internal	Pending
...			

5. If the remaining CSRs are not approved, and are in the **Pending** status, approve the CSRs for your cluster machines:

- To approve them individually, run the following command for each valid CSR:

```
$ oc adm certificate approve <csr_name>
```

where:

<csr_name>

Specifies the name of a CSR from the list of current CSRs.

- To approve all pending CSRs, run the following command:

```
$ oc get csr -o go-template='{{range .items}}{{if not .status}}{{.metadata.name}}\n{{end}}{{end}}' | xargs oc adm certificate approve
```

6. After all client and server CSRs have been approved, the machines have the **Ready** status.

Verify this by running the following command:

```
$ oc get nodes
```

Example output

NAME	STATUS	ROLES	AGE	VERSION
master-0	Ready	master	73m	v1.32.3
master-1	Ready	master	73m	v1.32.3
master-2	Ready	master	74m	v1.32.3
worker-0	Ready	worker	11m	v1.32.3
worker-1	Ready	worker	11m	v1.32.3

**NOTE**

It can take a few minutes after approval of the server CSRs for the machines to transition to the **Ready** status.

3.5.16. Initial Operator configuration

To ensure all Operators become available, configure the required Operators immediately after the control plane initialises. This configuration is essential for stabilizing the cluster environment following the installation.

Prerequisites

- Your control plane has initialized.

Procedure

1. Watch the cluster components come online:

```
$ watch -n5 oc get clusteroperators
```

Example output

NAME	VERSION	AVAILABLE	PROGRESSING	DEGRADED
SINCE				
authentication	4.19.0	True	False	False
baremetal	4.19.0	True	False	False
cloud-credential	4.19.0	True	False	False
cluster-autoscaler	4.19.0	True	False	False
config-operator	4.19.0	True	False	False

console	4.19.0	True	False	False	26m
csi-snapshot-controller	4.19.0	True	False	False	37m
dns	4.19.0	True	False	False	37m
etcd	4.19.0	True	False	False	36m
image-registry	4.19.0	True	False	False	31m
ingress	4.19.0	True	False	False	30m
insights	4.19.0	True	False	False	31m
kube-apiserver	4.19.0	True	False	False	26m
kube-controller-manager	4.19.0	True	False	False	36m
kube-scheduler	4.19.0	True	False	False	36m
kube-storage-version-migrator	4.19.0	True	False	False	37m
machine-api	4.19.0	True	False	False	29m
machine approver	4.19.0	True	False	False	37m
machine-config	4.19.0	True	False	False	36m
marketplace	4.19.0	True	False	False	37m
monitoring	4.19.0	True	False	False	29m
network	4.19.0	True	False	False	38m
node-tuning	4.19.0	True	False	False	37m
openshift-apiserver	4.19.0	True	False	False	32m
openshift-controller-manager	4.19.0	True	False	False	30m
openshift-samples	4.19.0	True	False	False	32m
operator-lifecycle-manager	4.19.0	True	False	False	37m
operator-lifecycle-manager-catalog	4.19.0	True	False	False	37m
operator-lifecycle-manager-packageserver	4.19.0	True	False	False	32m
service-ca	4.19.0	True	False	False	38m
storage	4.19.0	True	False	False	37m

2. Configure the Operators that are not available.

3.5.16.1. Disabling the default OperatorHub catalog sources

Operator catalogs that source content provided by Red Hat and community projects are configured for OperatorHub by default during an OpenShift Container Platform installation. In a restricted network environment, you must disable the default catalogs as a cluster administrator.

Procedure

- Disable the sources for the default catalogs by adding **disableAllDefaultSources: true** to the **OperatorHub** object:

```
$ oc patch OperatorHub cluster --type json \
  -p '[{"op": "add", "path": "/spec/disableAllDefaultSources", "value": true}]'
```

TIP

Alternatively, you can use the web console to manage catalog sources. From the **Administration** → **Cluster Settings** → **Configuration** → **OperatorHub** page, click the **Sources** tab, where you can create, update, delete, disable, and enable individual sources.

3.5.16.2. Image registry storage configuration

The Image Registry Operator is not initially available for platforms that do not provide default storage. After installation, you must configure your registry to use storage so that the Registry Operator is made available.

Configure a persistent volume, which is required for production clusters. Where applicable, you can configure an empty directory as the storage location for non-production clusters.

You can also allow the image registry to use block storage types by using the **Recreate** rollout strategy during upgrades.

3.5.16.2.1. Configuring registry storage for VMware vSphere

As a cluster administrator, following installation you must configure your registry to use storage.

Prerequisites

- Cluster administrator permissions.
- A cluster on VMware vSphere.
- Persistent storage provisioned for your cluster, such as Red Hat OpenShift Data Foundation.



IMPORTANT

OpenShift Container Platform supports **ReadWriteOnce** access for image registry storage when you have only one replica. **ReadWriteOnce** access also requires that the registry uses the **Recreate** rollout strategy. To deploy an image registry that supports high availability with two or more replicas, **ReadWriteMany** access is required.

- Must have "100Gi" capacity.



IMPORTANT

Testing shows issues with using the NFS server on RHEL as storage backend for core services. This includes the OpenShift Container Registry and Quay, Prometheus for monitoring storage, and Elasticsearch for logging storage. Therefore, using RHEL NFS to back PVs used by core services is not recommended.

Other NFS implementations on the marketplace might not have these issues. Contact the individual NFS implementation vendor for more information on any testing that was possibly completed against these OpenShift Container Platform core components.

Procedure

1. Change the **spec.storage.pvc** field in the **configs.imageregistry/cluster** resource.



NOTE

When you use shared storage, review your security settings to prevent outside access.

2. Verify that you do not have a registry pod by running the following command:

```
$ oc get pod -n openshift-image-registry -l docker-registry=default
```

Example output

No resources found in openshift-image-registry namespace



NOTE

If you do have a registry pod in your output, you do not need to continue with this procedure.

3. Check the registry configuration by running the following command:

```
$ oc edit configs.imageregistry.operator.openshift.io
```

Example output

```
storage:
pvc:
claim:
```

Leave the **claim** field blank to allow the automatic creation of an **image-registry-storage** persistent volume claim (PVC). The PVC is generated based on the default storage class. However, be aware that the default storage class might provide ReadWriteOnce (RWO) volumes, such as a RADOS Block Device (RBD), which can cause issues when you replicate to more than one replica.

4. Check the **clusteroperator** status by running the following command:

```
$ oc get clusteroperator image-registry
```

Example output

NAME	VERSION	AVAILABLE	PROGRESSING	DEGRADED	SINCE
MESSAGE					
image-registry	4.7	True	False	False	6h50m

3.5.16.2.2. Configuring storage for the image registry in non-production clusters

You must configure storage for the Image Registry Operator. For non-production clusters, you can set the image registry to an empty directory. If you do so, all images are lost if you restart the registry.

Procedure

- To set the image registry storage to an empty directory:

```
$ oc patch configs.imageregistry.operator.openshift.io cluster --type merge --patch '{"spec": {"storage":{"emptyDir":{}}}}'
```



WARNING

Configure this option only for non-production clusters.

If you run this command before the Image Registry Operator initializes its components, the **oc patch** command fails with the following error:

Error from server (NotFound): configs.imageregistry.operator.openshift.io "cluster" not found

Wait a few minutes and run the command again.

3.5.16.2.3. Configuring block registry storage for VMware vSphere

To allow the image registry to use block storage types such as vSphere Virtual Machine Disk (VMDK) during upgrades as a cluster administrator, you can use the **Recreate** rollout strategy.



IMPORTANT

Block storage volumes are supported but not recommended for use with image registry on production clusters. An installation where the registry is configured on block storage is not highly available because the registry cannot have more than one replica.

Procedure

1. Enter the following command to set the image registry storage as a block storage type, patch the registry so that it uses the **Recreate** rollout strategy, and runs with only **1** replica:

```
$ oc patch config.imageregistry.operator.openshift.io/cluster --type=merge -p '{"spec": {"rolloutStrategy":"Recreate", "replicas":1}}'
```

2. Provision the persistent volume (PV) for the block storage device, and create a persistent volume claim (PVC) for that volume. The requested block volume uses the ReadWriteOnce (RWO) access mode.
 - a. Create a **pvc.yaml** file with the following contents to define a VMware vSphere **PersistentVolumeClaim** object:

```
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: image-registry-storage
  namespace: openshift-image-registry
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 100Gi
```

where:

name

Specifies a unique name that represents the **PersistentVolumeClaim** object.

namespace

Specifies the **namespace** for the **PersistentVolumeClaim** object, which is **openshift-image-registry**.

accessModes

Specifies the access mode of the persistent volume claim. With **ReadWriteOnce**, the volume can be mounted with read and write permissions by a single node.

storage

The size of the persistent volume claim.

- b. Enter the following command to create the **PersistentVolumeClaim** object from the file:

```
$ oc create -f pvc.yaml -n openshift-image-registry
```

3. Enter the following command to edit the registry configuration so that it references the correct PVC:

```
$ oc edit config.imageregistry.operator.openshift.io -o yaml
```

Example output

```
storage:
pvc:
claim:
```

By creating a custom PVC, you can leave the **claim** field blank for the default automatic creation of an **image-registry-storage** PVC.

For instructions about configuring registry storage so that it references the correct PVC, see [Configuring the registry for vSphere](#).

3.5.17. Completing installation on user-provisioned infrastructure

To finalize the installation on user-provisioned infrastructure, complete the cluster deployment after configuring the Operators. This ensures the cluster is fully operational on the infrastructure that you provide.

Prerequisites

- Your control plane has initialized.
- You have completed the initial Operator configuration.

Procedure

1. Confirm that all the cluster components are online with the following command:

```
$ watch -n5 oc get clusteroperators
```

Example output

NAME	VERSION	AVAILABLE	PROGRESSING	DEGRADED
SINCE				
authentication	4.19.0	True	False	False 19m
baremetal	4.19.0	True	False	False 37m
cloud-credential	4.19.0	True	False	False 40m
cluster-autoscaler	4.19.0	True	False	False 37m
config-operator	4.19.0	True	False	False 38m
console	4.19.0	True	False	False 26m
csi-snapshot-controller	4.19.0	True	False	False 37m
dns	4.19.0	True	False	False 37m
etcd	4.19.0	True	False	False 36m
image-registry	4.19.0	True	False	False 31m
ingress	4.19.0	True	False	False 30m
insights	4.19.0	True	False	False 31m
kube-apiserver	4.19.0	True	False	False 26m
kube-controller-manager	4.19.0	True	False	False 36m
kube-scheduler	4.19.0	True	False	False 36m
kube-storage-version-migrator	4.19.0	True	False	False 37m
machine-api	4.19.0	True	False	False 29m
machine approver	4.19.0	True	False	False 37m
machine-config	4.19.0	True	False	False 36m
marketplace	4.19.0	True	False	False 37m
monitoring	4.19.0	True	False	False 29m
network	4.19.0	True	False	False 38m
node-tuning	4.19.0	True	False	False 37m
openshift-apiserver	4.19.0	True	False	False 32m
openshift-controller-manager	4.19.0	True	False	False 30m
openshift-samples	4.19.0	True	False	False 32m
operator-lifecycle-manager	4.19.0	True	False	False 37m
operator-lifecycle-manager-catalog	4.19.0	True	False	False 37m
operator-lifecycle-manager-packageserver	4.19.0	True	False	False 32m
service-ca	4.19.0	True	False	False 38m
storage	4.19.0	True	False	False 37m

Alternatively, the following command notifies you when all of the clusters are available. The command also retrieves and displays credentials:

```
$ ./openshift-install --dir <installation_directory> wait-for install-complete
```

where:

<installation_directory>

Specifies the path to the directory that you stored the installation files in.

Example output

```
INFO Waiting up to 30m0s for the cluster to initialize...
```

The command succeeds when the Cluster Version Operator finishes deploying the OpenShift Container Platform cluster from Kubernetes API server.



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.
- It is recommended that you use 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.

2. Confirm that the Kubernetes API server is communicating with the pods.

a. To view a list of all pods, use the following command:

```
$ oc get pods --all-namespaces
```

Example output

NAMESPACE	NAME	READY	STATUS
RESTARTS	AGE		
openshift-apiserver-operator	openshift-apiserver-operator-85cb746d55-zqhs8	1/1	
Running 1 9m			
openshift-apiserver	apiserver-67b9g	1/1	Running 0
3m			
openshift-apiserver	apiserver-ljcmx	1/1	Running 0
1m			
openshift-apiserver	apiserver-z25h4	1/1	Running 0
2m			
openshift-authentication-operator	authentication-operator-69d5d8bf84-vh2n8	1/1	
Running 0 5m			

b. View the logs for a pod that is listed in the output of the previous command by using the following command:

```
$ oc logs <pod_name> -n <namespace>
```

where:

<namespace>

Specifies the pod name and namespace, as shown in the output of an earlier command. If the pod logs display, the Kubernetes API server can communicate with the cluster machines.

3. For an installation with Fibre Channel Protocol (FCP), additional steps are required to enable multipathing. Do not enable multipathing during installation. See "Enabling multipathing with kernel arguments on RHCOS" in the *Postinstallation machine configuration tasks* documentation for more information.
4. Register your cluster on the [Cluster registration](#) page.

You can add extra compute machines after the cluster installation is completed by following [Adding compute machines to vSphere](#).

3.5.18. Configuring vSphere DRS anti-affinity rules for control plane nodes

vSphere Distributed Resource Scheduler (DRS) anti-affinity rules can be configured to support higher availability of OpenShift Container Platform Control Plane nodes. Anti-affinity rules ensure that the vSphere Virtual Machines for the OpenShift Container Platform Control Plane nodes are not scheduled to the same vSphere Host.



IMPORTANT

- The following information applies to compute DRS only and does not apply to storage DRS.
- The **govc** command is an open-source command available from VMware; it is not available from Red Hat. The **govc** command is not supported by the Red Hat support.
- Instructions for downloading and installing **govc** are found on the VMware documentation website.

Create an anti-affinity rule by running the following command:

Example command

```
$ govc cluster.rule.create \
-name openshift4-control-plane-group \
-dc MyDatacenter -cluster MyCluster \
-enable \
-anti-affinity master-0 master-1 master-2
```

After creating the rule, your control plane nodes are automatically migrated by vSphere so they are not running on the same hosts. This might take some time while vSphere reconciles the new rule. Successful command completion is shown in the following procedure.



NOTE

The migration occurs automatically and might cause brief OpenShift API outage or latency until the migration finishes.

The vSphere DRS anti-affinity rules need to be updated manually in the event of a control plane VM name change or migration to a new vSphere Cluster.

Procedure

1. Remove any existing DRS anti-affinity rule by running the following command:

```
$ govc cluster.rule.remove \
-name openshift4-control-plane-group \
-dc MyDatacenter -cluster MyCluster
```

Example Output

```
[13-10-22 09:33:24] Reconfigure /MyDatacenter/host/MyCluster...OK
```

2. Create the rule again with updated names by running the following command:

```
$ govc cluster.rule.create \
-name openshift4-control-plane-group \
-dc MyDatacenter -cluster MyOtherCluster \
-enable \
-anti-affinity master-0 master-1 master-2
```

3.5.19. Telemetry access for OpenShift Container Platform

To provide metrics about cluster health and the success of updates, the Telemetry service requires internet access. When connected, this service runs automatically by default and registers your cluster to [OpenShift Cluster Manager](#).

After you confirm that your [OpenShift Cluster Manager](#) inventory is correct, either maintained automatically by Telemetry or manually by using OpenShift Cluster Manager, use subscription watch to track your OpenShift Container Platform subscriptions at the account or multi-cluster level. For more information about subscription watch, see "Data Gathered and Used by Red Hat's subscription services" in the *Additional resources* section.

Additional resources

- See [About remote health monitoring](#) for more information about the Telemetry service

3.5.20. Next steps

- [Customize your cluster](#).
- If the mirror registry that you used to install your cluster has a trusted CA, add it to the cluster by [configuring additional trust stores](#).
- If necessary, you can [Remote health reporting](#).
- Optional: [View the events from the vSphere Problem Detector Operator](#) to determine if the cluster has permission or storage configuration issues.
- Optional: if you created encrypted virtual machines, [create an encrypted storage class](#).

CHAPTER 4. INSTALLING A CLUSTER ON VSphere USING THE ASSISTED INSTALLER

You can install OpenShift Container Platform on on-premise hardware or on-premise VMs by using the Assisted Installer. Installing OpenShift Container Platform by using the Assisted Installer supports **x86_64**, **AArch64**, **ppc64le**, and **s390x** CPU architectures.

The Assisted Installer is a user-friendly installation solution offered on the Red Hat Hybrid Cloud Console.

4.1. ADDITIONAL RESOURCES

- [Installing OpenShift Container Platform with the Assisted Installer](#)

CHAPTER 5. INSTALLING A CLUSTER ON VSHERE USING THE AGENT-BASED INSTALLER

The Agent-based installation method provides the flexibility to boot your on-premise servers in any way that you choose. It combines the ease of use of the Assisted Installation service with the ability to run offline, including in air-gapped environments.

Agent-based installation is a subcommand of the OpenShift Container Platform installer. It generates a bootable ISO image containing all of the information required to deploy an OpenShift Container Platform cluster with an available release image.

For more information about installing a cluster using the Agent-based Installer, see [Preparing to install with the Agent-based Installer](#).



IMPORTANT

Your vSphere account must include privileges for reading and creating the resources required to install an OpenShift Container Platform cluster. For more information about privileges, see [vCenter requirements](#).

CHAPTER 6. INSTALLING A THREE-NODE CLUSTER ON VSphere

In OpenShift Container Platform version 4.19, you can install a three-node cluster on VMware vSphere. A three-node cluster consists of three control plane machines, which also act as compute machines. This type of cluster provides a smaller, more resource efficient cluster, for cluster administrators and developers to use for testing, development, and production.

You can install a three-node cluster using either installer-provisioned or user-provisioned infrastructure.

6.1. CONFIGURING A THREE-NODE CLUSTER

To configure a three-node cluster, set the number of worker nodes to **0** in the **install-config.yaml** file before you deploy the cluster.

Setting the number of worker nodes to **0** ensures that the control plane machines are schedulable. This allows application workloads to be scheduled to run from the control plane nodes.



NOTE

Because application workloads run from control plane nodes, additional subscriptions are required, as the control plane nodes are considered to be compute nodes.

Prerequisites

- You have an existing **install-config.yaml** file.

Procedure

1. Set the number of compute replicas to **0** in your **install-config.yaml** file, as shown in the following **compute** stanza:

Example **install-config.yaml** file for a three-node cluster

```
apiVersion: v1
baseDomain: example.com
compute:
- name: worker
  platform: {}
  replicas: 0
# ...
```

2. If you are deploying a cluster with user-provisioned infrastructure:

- Configure your application ingress load balancer to route HTTP and HTTPS traffic to the control plane nodes. In a three-node cluster, the Ingress Controller pods run on the control plane nodes. For more information, see the "Load balancing requirements for user-provisioned infrastructure".
- After you create the Kubernetes manifest files, make sure that the **spec.mastersScheduleable** parameter is set to **true** in **cluster-scheduler-02-config.yaml** file. You can locate this file in **<installation_directory>/manifests**. For more information,

see "Creating the Kubernetes manifest and Ignition config files" in "Installing a cluster on vSphere with user-provisioned infrastructure".

- Do not create additional worker nodes.

Example `cluster-scheduler-02-config.yml` file for a three-node cluster

```
apiVersion: config.openshift.io/v1
kind: Scheduler
metadata:
  creationTimestamp: null
  name: cluster
spec:
  mastersSchedulable: true
  policy:
    name: ""
  status: {}
```

6.2. NEXT STEPS

- [Installing a cluster on vSphere with customizations](#)
- [Installing a cluster on vSphere with user-provisioned infrastructure](#)

CHAPTER 7. UNINSTALLING A CLUSTER ON VSphere THAT USES INSTALLER-PROVISIONED INFRASTRUCTURE

You can remove a cluster that you deployed in your VMware vSphere instance by using installer-provisioned infrastructure.

7.1. REMOVING A CLUSTER THAT USES INSTALLER-PROVISIONED INFRASTRUCTURE

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



NOTE

After uninstallation, check your cloud provider for any resources that were not removed properly, especially with user-provisioned infrastructure clusters. Some resources might exist because either the installation program did not create the resource or could not access the resource.

Prerequisites

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

Procedure

1. From the directory that has the installation program on the computer that you used to install the cluster, run the following command:

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

where:

<installation_directory>

Specify the path to the directory that you stored the installation files in.

--log-level info

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



NOTE

You must specify the directory that includes 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.

CHAPTER 8. USING THE VSphere PROBLEM DETECTOR OPERATOR

You can use the vSphere Problem Detector Operator to check a cluster that you deployed on VMware vSphere for common installation and misconfiguration issues that relate to storage.

8.1. ABOUT THE VSphere PROBLEM DETECTOR OPERATOR

The vSphere Problem Detector Operator checks a cluster that you deployed on VMware vSphere for common installation and configuration issues that relate to storage.

After the Cluster Storage Operator starts and determines that a cluster runs on VMware vSphere, the Cluster Storage Operator launches the vSphere Problem Detector Operator. When the vSphere Problem Detector Operator starts, the Operator immediately runs the checks. The vSphere Problem Detector Operator communicates with the vSphere vCenter Server to find the virtual machines in the cluster, the default datastore, and other information about the vSphere vCenter Server configuration. The Operator uses the credentials from the Cloud Credential Operator to connect to vSphere.

The Operator runs the checks according to the following schedule:

- The checks run every hour.
- If any check fails, the Operator runs the checks again in intervals of 1 minute, 2 minutes, 4, 8, and so on. The Operator doubles the interval up to a maximum interval of 8 hours.
- When all checks pass, the schedule returns to an hour interval.

After a failure, the Operator increases its check frequency to quickly report success when the failure condition gets resolved. You can run the Operator manually for immediate troubleshooting information.

8.2. RUNNING THE VSphere PROBLEM DETECTOR OPERATOR CHECKS

You can override the schedule for running the vSphere Problem Detector Operator checks and run the checks immediately.

The vSphere Problem Detector Operator automatically runs the checks every hour. After the Operator starts, the Operator runs the checks immediately. After the Cluster Storage Operator starts and determines that a cluster runs on VMware vSphere, the Cluster Storage Operator starts the vSphere Problem Detector Operator. To run the checks immediately, you can scale the vSphere Problem Detector Operator to **0** and back to **1** so that the Cluster Storage Operator restarts the vSphere Problem Detector Operator.

Prerequisites

- Access to the cluster as a user with the **cluster-admin** role.

Procedure

- Scale the Operator to **0**:

```
$ oc scale deployment/vsphere-problem-detector-operator --replicas=0 \
-n openshift-cluster-storage-operator
```

Verification

- Verify that the pods have restarted by running the following command:

```
$ oc -n openshift-cluster-storage-operator get pod -l name=vsphere-problem-detector-operator -w
```

Example output

NAME	READY	STATUS	RESTARTS	AGE
vsphere-problem-detector-operator-77486bd645-9ntpb	1/1	Running	0	11s

The **AGE** field must indicate that the pod restarted.

Next steps

- Viewing the events from the vSphere Problem Detector Operator
- Viewing the logs from the vSphere Problem Detector Operator

8.3. VIEWING THE EVENTS FROM THE VSphere PROBLEM DETECTOR OPERATOR

After the vSphere Problem Detector Operator runs and performs the configuration checks, the Operator creates events that you can view from the command-line interface (CLI) or from the OpenShift Container Platform web console.

Prerequisites

- The vSphere Problem Detector Operator ran checks on your cluster.

Procedure

- To view the events by using the CLI, run the following command:

```
$ oc get event -n openshift-cluster-storage-operator \
--sort-by={.metadata.creationTimestamp}
```

Example output

16m	Normal	Started	pod/vsphere-problem-detector-operator-xxxxx	Started
			container vsphere-problem-detector	
16m	Normal	Created	pod/vsphere-problem-detector-operator-xxxxx	Created
			container vsphere-problem-detector	
16m	Normal	LeaderElection	configmap/vsphere-problem-detector-lock	vsphere-problem-detector-operator-xxxxx became leader

- To view the events by using the OpenShift Container Platform web console, navigate to **Home** → **Events** and select **openshift-cluster-storage-operator** from the **Project** menu.

8.4. VIEWING THE LOGS FROM THE VSphere PROBLEM DETECTOR OPERATOR

After the vSphere Problem Detector Operator runs and performs the configuration checks, the Operator creates log records that you can view from the command-line interface (CLI) or from the OpenShift Container Platform web console. Log lines that indicate **passed** means that you do not need to perform any actions.

The ideal output for a log line indicates **passed** or **0 problems**. If a log line indicates **failure** or 1 or more problems, see the information in the "Configuration checks run by the vSphere Problem Detector Operator" document.

Prerequisites

- The vSphere Problem Detector Operator ran checks on your cluster.

Procedure

- To view the logs by using the CLI, run the following command. A log line that shows **passed** in the output means that you must analyze the log output and resolve the issue.

```
$ oc logs deployment/vsphere-problem-detector-operator \
-n openshift-cluster-storage-operator
```

Example output

```
I0108 08:32:28.445696    1 operator.go:209] ClusterInfo passed
I0108 08:32:28.451029    1 datastore.go:57] CheckStorageClasses checked 1 storage
classes, 0 problems found
I0108 08:32:28.451047    1 operator.go:209] CheckStorageClasses passed
I0108 08:32:28.452160    1 operator.go:209] CheckDefaultDatastore passed
I0108 08:32:28.480648    1 operator.go:271] CheckNodeDiskUUID:<host_name> passed
I0108 08:32:28.480685    1 operator.go:271] CheckNodeProviderID:<host_name> passed
```

- To view the Operator logs with the OpenShift Container Platform web console, perform the following steps:
 - a. Navigate to **Workloads** → **Pods**.
 - b. Select **openshift-cluster-storage-operator** from the **Projects** menu.
 - c. Click the link for the **vsphere-problem-detector-operator** pod.
 - d. Click the **Logs** tab on the **Pod details** page to view the logs.

8.5. CONFIGURATION CHECKS RUN BY THE VSphere PROBLEM DETECTOR OPERATOR

The following tables identify the configuration checks that the vSphere Problem Detector Operator runs. Some checks verify the configuration of the cluster. Other checks verify the configuration of each node in the cluster.

Table 8.1. Cluster configuration checks

Name	Description
CheckDefaultDatastore	<p>Verifies that the default datastore name in the VMware vSphere configuration is short enough for use with dynamic provisioning.</p> <p>If this check fails, you can expect the following:</p> <ul style="list-style-type: none"> ● systemd logs errors to the journal such as Failed to set up mount unit: Invalid argument. ● systemd does not unmount volumes if the virtual machine shuts down or reboots without draining all the pods from the node. <p>If this check fails, reconfigure vSphere with a shorter name for the default datastore.</p>
CheckFolderPermissions	<p>Verifies the permission to list volumes in the default datastore. You must enable the permission to create volumes. The Operator verifies the permission by listing the <code>/</code> and <code>/kubevols</code> directories. When the Operator performs the check, the root directory must exist. The <code>/kubevols</code> directory might not exist at the time of the check. The creation of the <code>/kubevols</code> directory occurs when the datastore supports dynamic provisioning.</p> <p>If this check fails, review the required permissions for the vCenter account that you specified during the OpenShift Container Platform installation.</p>
CheckStorageClasses	<p>Verifies the following:</p> <ul style="list-style-type: none"> ● The fully qualified path to each persistent volume that the storage class provisions does not go lower than 255 characters. ● The storage class can use only one storage policy and the policy must be defined.
CheckTaskPermissions	Verifies the permission to list recent tasks and datastores.
ClusterInfo	Collects the cluster version and UUID from vSphere vCenter.

Table 8.2. Node configuration checks

Name	Description
CheckNodeDiskUUID	<p>Verifies that all the vSphere virtual machines include the disk.enableUUID=TRUE configuration.</p> <p>If this check fails, see the How to check <code>disk.EnableUUID</code> parameter from VM in vSphere Red Hat Knowledgebase solution.</p>

Name	Description
CheckNodeProviderID	<p>Verifies that all nodes have the ProviderID configuration from vSphere vCenter. This check fails when the output from the following command does not include a provider ID for each node.</p> <pre>\$ oc get nodes -o custom-columns=NAME:.metadata.name,PROVIDER_ID:.spec.providerID,UUID:.status.nodeInfo.systemUUID</pre> <p>If this check fails, reference the vSphere product documentation on how to set the provider ID for each node in the cluster.</p>
CollectNodeESXiVersion	Reports the version of the ESXi hosts that run nodes.
CollectNodeHWVersion	Reports the virtual machine hardware version for a node.

8.6. ABOUT THE STORAGE CLASS CONFIGURATION CHECK

The datastore name and cluster ID relate to the names for persistent volumes that use VMware vSphere storage. After the creation of a persistent volume, **systemd** creates a mount unit for the persistent volume.

The **systemd** process has a 255 character limit for the length of the fully qualified path to the virtual machine disk (VMDK) file. This path follows the naming conventions for **systemd** and vSphere. The naming conventions use the following example pattern:

```
/var/lib/kubelet/plugins/kubernetes.io/vsphere-volume/mounts/[<datastore>] 00000000-0000-0000-0000-000000000000/<cluster_id>-dynamic-pvc-00000000-0000-0000-000000000000.vmdk
```

- The naming conventions require 205 characters of the 255 character limit.
- The deployment determines the datastore name and the cluster ID.
- The datastore name and cluster ID substitute into the example pattern. The fully qualified path gets processed with the **systemd-escape** command to escape special characters. For example, after the escape operation, a hyphen character uses four characters, such as **\x2d**.
- After the **systemd-escape** CLI processes the VMDK file path, the length of the path must not be lower than 255 characters. This criteria ensures that the **systemd** process can access the fully qualified VMDK file path.

8.7. METRICS FOR THE VSphere PROBLEM DETECTOR OPERATOR

The vSphere Problem Detector Operator exposes the following metrics for use by the OpenShift Container Platform monitoring stack.

Table 8.3. Metrics exposed by the vSphere Problem Detector Operator

Name	Description
vsphere_cluster_check_total	Cumulative number of cluster-level checks that the vSphere Problem Detector Operator performed. This count includes both successes and failures.
vsphere_cluster_check_errors	Number of failed cluster-level checks that the vSphere Problem Detector Operator performed. For example, a value of 1 indicates that one cluster-level check failed.
vsphere_esxi_version_total	Counts the number of ESXi hosts with a specific version. Note that if a host runs more than one node, the vSphere Problem Detector Operator counts the host only once.
vsphere_node_check_total	Cumulative number of node-level checks that the vSphere Problem Detector Operator performed. This count includes both successes and failures.
vsphere_node_check_errors	Counts the number of failed node-level checks that the vSphere Problem Detector Operator performed. For example, a value of 1 indicates that one node-level check failed.
vsphere_node_hw_version_total	Number of vSphere nodes with a specific hardware version.
vsphere_vcenter_info	Information about the vSphere vCenter Server.

8.8. ADDITIONAL RESOURCES

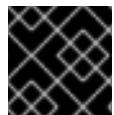
- [About OpenShift Container Platform monitoring](#)

CHAPTER 9. INSTALLATION CONFIGURATION PARAMETERS FOR VSphere

Before you deploy an OpenShift Container Platform cluster on vSphere, you can configure parameters to customize your cluster and the platform that hosts it. The installation program uses the information in the **install-config.yaml** file to provision required infrastructure and deploy cluster components. When you create the **install-config.yaml** file, you can configure the values for your required parameters through the command line. Edit the **install-config.yaml** file to customize your cluster further before installation begins.

9.1. AVAILABLE INSTALLATION CONFIGURATION PARAMETERS FOR VSphere

The following tables specify the required, optional, and vSphere-specific installation configuration parameters that you can set as part of the installation process.



IMPORTANT

After installation, you cannot change these parameters in the **install-config.yaml** file.

9.1.1. Required configuration parameters

Required installation configuration parameters are described in the following table:

Table 9.1. Required parameters

Parameter	Description
apiVersion:	<p>The API version for the install-config.yaml content. The current version is v1. The installation program might also support older API versions.</p> <p>Value: String</p>
baseDomain:	<p>The base domain of your cloud provider. The base domain 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.</p> <p>Value: A fully-qualified domain or subdomain name, such as example.com.</p>
metadata:	<p>Kubernetes resource ObjectMeta, from which only the name parameter is consumed.</p> <p>Value: Object</p>

Parameter	Description
metadata: name:	<p>The name of the cluster. DNS records for the cluster are all subdomains of {{.metadata.name}} and {{.baseDomain}}.</p> <p>Value: String of lowercase letters and hyphens (-), such as dev.</p>
platform:	<p>The configuration for the specific platform upon which to perform the installation: aws, baremetal, azure, gcp, ibmcloud, nutanix, openstack, powervs, vsphere, or {}. For additional information about platform.<platform> parameters, consult the table for your specific platform that follows.</p> <p>Value: Object</p>
pullSecret:	<p>Get a pull secret from Red Hat OpenShift Cluster Manager to authenticate downloading container images for OpenShift Container Platform components from services such as Quay.io.</p> <p>Value:</p> <pre data-bbox="817 1156 1285 1605"> { "auths": { "cloud.openshift.com": { "auth": "b3Blb=", "email": "you@example.com" }, "quay.io": { "auth": "b3Blb=", "email": "you@example.com" } } }</pre>

9.1.2. Network configuration parameters

You can customize your installation configuration based on the requirements of your existing network infrastructure. For example, you can expand the IP address block for the cluster network or configure different IP address blocks than the defaults.

Consider the following information before you configure network parameters for your cluster:

- If you use the Red Hat OpenShift Networking OVN-Kubernetes network plugin, both IPv4 and IPv6 address families are supported.

- If you deployed nodes in an OpenShift Container Platform cluster with a network that supports both IPv4 and non-link-local IPv6 addresses, configure your cluster to use a dual-stack network.
 - For clusters configured for dual-stack networking, both IPv4 and IPv6 traffic must use the same network interface as the default gateway. This ensures that in a multiple network interface controller (NIC) environment, a cluster can detect what NIC to use based on the available network interface. For more information, see "OVN-Kubernetes IPv6 and dual-stack limitations" in *About the OVN-Kubernetes network plugin*.
 - To prevent network connectivity issues, do not install a single-stack IPv4 cluster on a host that supports dual-stack networking.



NOTE

On VMware vSphere, dual-stack networking can specify either IPv4 or IPv6 as the primary address family.

If you configure your cluster to use both IP address families, review the following requirements:

- Both IP families must use the same network interface for the default gateway.
- Both IP families must have the default gateway.
- You must specify IPv4 and IPv6 addresses in the same order for all network configuration parameters. For example, in the following configuration, IPv4 addresses are listed before IPv6 addresses:

```
networking:
  clusterNetwork:
    - cidr: 10.128.0.0/14
      hostPrefix: 23
    - cidr: fd00:10:128::/56
      hostPrefix: 64
  serviceNetwork:
    - 172.30.0.0/16
    - fd00:172:16::/112
```

Table 9.2. Network parameters

Parameter	Description
<p>networking:</p>	<p>The configuration for the cluster network.</p> <p>Value: Object</p> <div data-bbox="809 1763 912 1920" style="float: right;"> </div> <div data-bbox="991 1763 1087 1801" style="float: right;"> NOTE </div> <p>You cannot change parameters specified by the networking object after installation.</p>

Parameter	Description
networking: networkType:	<p>The Red Hat OpenShift Networking network plugin to install.</p> <p>Value: OVNKubernetes. OVNKubernetes is a Container Network Interface (CNI) plugin for Linux networks and hybrid networks that contain both Linux and Windows servers. The default value is OVNKubernetes.</p>
networking: clusterNetwork:	<p>The IP address blocks for pods. The default value is 10.128.0.0/14 with a host prefix of /23.</p> <p>If you specify multiple IP address blocks, the blocks must not overlap.</p> <p>Value: An array of objects. For example:</p> <pre data-bbox="825 898 1126 1055">networking: clusterNetwork: - cidr: 10.128.0.0/14 hostPrefix: 23</pre>
networking: clusterNetwork: cidr:	<p>Required if you use networking.clusterNetwork. An IP address block. An IPv4 network.</p> <p>Value: An IP address block in Classless Inter-Domain Routing (CIDR) notation. The prefix length for an IPv4 block is between 0 and 32.</p>
networking: clusterNetwork: hostPrefix:	<p>The subnet prefix length to assign to each individual node. For example, if hostPrefix is set to 23 then each node is assigned a /23 subnet out of the given cidr. A hostPrefix value of 23 provides 510 ($2^{32} - 23 - 2$) pod IP addresses.</p> <p>Value: A subnet prefix. The default value is 23.</p>

Parameter	Description
<pre data-bbox="165 256 425 323">networking: serviceNetwork:</pre>	<p>The IP address block for services. The default value is 172.30.0.0/16. The OVN-Kubernetes network plugins supports only a single IP address block for the service network. Value: An array with an IP address block in CIDR format. For example:</p> <pre data-bbox="850 541 1075 642">networking: serviceNetwork: - 172.30.0.0/16</pre>
<pre data-bbox="196 750 438 817">networking: machineNetwork:</pre>	<p>The IP address blocks for machines. If you specify multiple IP address blocks, the blocks must not overlap. Value: An array of objects. For example:</p> <pre data-bbox="850 956 1091 1057">networking: machineNetwork: - cidr: 10.0.0.0/16</pre>
<pre data-bbox="196 1172 438 1273">networking: machineNetwork: cidr:</pre>	<p>Required if you use networking.machineNetwork. An IP address block. The default value is 10.0.0.0/16 for all platforms other than libvirt and IBM Power® Virtual Server. For libvirt, the default value is 192.168.126.0/24. For IBM Power® Virtual Server, the default value is 192.168.0.0/24. Value: An IP network block in CIDR notation. For example, 10.0.0.0/16.</p> <div data-bbox="810 1567 917 1769" data-label="Image"> </div> <p data-bbox="990 1567 1091 1603">NOTE</p> <p data-bbox="990 1637 1421 1769">Set the networking.machineNetwork to match the CIDR that the preferred NIC resides in.</p>

Parameter	Description
networking: ovnKubernetesConfig: ipv4: internalJoinSubnet:	Configures the IPv4 join subnet that is used internally by ovn-kubernetes . This subnet must not overlap with any other subnet that OpenShift Container Platform is using, including the node network. The size of the subnet must be larger than the number of nodes. You cannot change the value after installation. Value: An IP network block in CIDR notation. The default value is 100.64.0.0/16 .

9.1.3. Optional configuration parameters

Optional installation configuration parameters are described in the following table:

Table 9.3. Optional parameters

Parameter	Description
additionalTrustBundle:	A PEM-encoded X.509 certificate bundle that is added to the nodes' trusted certificate store. This trust bundle might also be used when a proxy has been configured. Value: String
capabilities:	Controls the installation of optional core cluster components. You can reduce the footprint of your OpenShift Container Platform cluster by disabling optional components. For more information, see the "Cluster capabilities" page in <i>Installing</i> . Value: String array
capabilities: baselineCapabilitySet:	Selects an initial set of optional capabilities to enable. Valid values are None , v4.11 , v4.12 and vCurrent . The default value is vCurrent . Value: String
capabilities: additionalEnabledCapabilities:	Extends the set of optional capabilities beyond what you specify in baselineCapabilitySet . You can specify multiple capabilities in this parameter. Value: String array

Parameter	Description
cpuPartitioningMode:	<p>Enables workload partitioning, which isolates OpenShift Container Platform services, cluster management workloads, and infrastructure pods to run on a reserved set of CPUs. You can only enable workload partitioning during installation. You cannot disable it after installation. While this field enables workload partitioning, it does not configure workloads to use specific CPUs. For more information, see the <i>Workload partitioning</i> page in the <i>Scalability and Performance</i> section.</p> <p>Value: None or AllNodes. None is the default value.</p>
compute:	<p>The configuration for the machines that comprise the compute nodes.</p> <p>Value: Array of MachinePool objects.</p>
compute: architecture:	<p>Determines the instruction set architecture of the machines in the pool. Currently, clusters with varied architectures are not supported. All pools must specify the same architecture. Valid values are amd64 (the default).</p> <p>Value: String</p>
compute: name:	<p>Required if you use compute. The name of the machine pool.</p> <p>Value: worker</p>
compute: platform:	<p>Required if you use compute. Use this parameter to specify the cloud provider to host the worker machines. This parameter value must match the controlPlane.platform parameter value.</p> <p>Value: aws, azure, gcp, ibmcloud, nutanix, openstack, powervs, vsphere, or {}</p>
compute: replicas:	<p>The number of compute machines, which are also known as worker machines, to provision.</p> <p>Value: A positive integer greater than or equal to 2. The default value is 3.</p>

Parameter	Description
featureSet:	<p>Enables the cluster for a feature set. A feature set is a collection of OpenShift Container Platform features that are not enabled by default. For more information about enabling a feature set during installation, see "Enabling features using feature gates".</p> <p>Value: String. The name of the feature set to enable, such as TechPreviewNoUpgrade.</p>
controlPlane:	<p>The configuration for the machines that form the control plane.</p> <p>Value: Array of MachinePool objects.</p>
controlPlane: architecture:	<p>Determines the instruction set architecture of the machines in the pool. Currently, clusters with varied architectures are not supported. All pools must specify the same architecture. Valid values are amd64 (the default).</p> <p>Value: String</p>
controlPlane: name:	<p>Required if you use controlPlane. The name of the machine pool.</p> <p>Value: master</p>
controlPlane: platform:	<p>Required if you use controlPlane. Use this parameter to specify the cloud provider that hosts the control plane machines. This parameter value must match the compute.platform parameter value.</p> <p>Value: aws, azure, gcp, ibmcloud, nutanix, openstack, powervs, vsphere, or {}</p>
controlPlane: replicas:	<p>The number of control plane machines to provision.</p> <p>Value: Supported values are 3, or 1 when deploying single-node OpenShift.</p>

Parameter	Description
<p>credentialsMode:</p>	<p>The Cloud Credential Operator (CCO) mode. If no mode is specified, the CCO dynamically tries to determine the capabilities of the provided credentials, with a preference for mint mode on the platforms where multiple modes are supported.</p> <p></p> <p>NOTE</p> <p>Not all CCO modes are supported for all cloud providers. For more information about CCO modes, see the "Managing cloud provider credentials" entry in the <i>Authentication and authorization</i> content.</p> <p>Value: Mint, Passthrough, Manual or an empty string ("").</p>

Parameter	Description
<p>tips:</p>	<p>Enable or disable FIPS mode. The default is false (disabled). If you enable FIPS mode, 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 RHCOS provides instead.</p> <p>IMPORTANT</p> <p>To enable FIPS mode for your cluster, you must run the installation program from a Red Hat Enterprise Linux (RHEL) computer configured to operate in FIPS mode. For more information about configuring FIPS mode on RHEL, see Switching RHEL to FIPS mode.</p> <p>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.</p>
	<p>IMPORTANT</p> <p>If you are using Azure File storage, you cannot enable FIPS mode.</p>
	<p>Value: false or true</p>
<p>imageContentSources:</p>	<p>Sources and repositories for the release-image content.</p> <p>Value: Array of objects. Includes asource and, optionally, mirrors, as described in the following rows of this table.</p>
<p>imageContentSources: source:</p>	<p>Required if you use imageContentSources. Specify the repository that users refer to, for example, in image pull specifications.</p> <p>Value: String</p>
<p>imageContentSources: mirrors:</p>	<p>Specify one or more repositories that might also contain the same images.</p> <p>Value: Array of strings</p>

Parameter	Description
<p>publish:</p>	<p>How to publish or expose the user-facing endpoints of your cluster, such as the Kubernetes API, OpenShift routes.</p> <p>Value: Internal or External. The default value is External.</p> <p>Setting this field to Internal is not supported on non-cloud platforms.</p> <p></p> <p>IMPORTANT</p> <p>If the value of the field is set to Internal, the cluster becomes non-functional. For more information, refer to BZ#1953035.</p>
<p>sshKey:</p>	<p>The SSH key to authenticate access to your cluster machines.</p> <p></p> <p>NOTE</p> <p>For production OpenShift Container Platform clusters on which you want to perform installation debugging or disaster recovery, specify an SSH key that your ssh-agent process uses.</p> <p>Value: For example, sshKey: ssh-ed25519 AAAA...</p>

9.1.4. Additional VMware vSphere configuration parameters

Additional VMware vSphere configuration parameters are described in the following table:

Table 9.4. Additional VMware vSphere cluster parameters

Parameter	Description
<p>platform: vsphere:</p>	<p>Describes your account on the cloud platform that hosts your cluster. You can use the parameter to customize the platform. If you provide additional configuration settings for compute and control plane machines in the machine pool, the parameter is not required.</p> <p>Value: A dictionary of vSphere configuration objects</p>

Parameter	Description
<pre>platform: vsphere: apiVIPs:</pre>	<p>Virtual IP (VIP) addresses that you configured for control plane API access.</p> <p> NOTE</p> <p>This parameter applies only to installer-provisioned infrastructure without an external load balancer configured. You must not specify this parameter in user-provisioned infrastructure.</p> <p>Value: Multiple IP addresses</p>
<pre>platform: vsphere: diskType:</pre>	<p>Optional: The disk provisioning method. This value defaults to the vSphere default storage policy if not set.</p> <p>Value: Valid values are thin, thick, or eagerZeroedThick.</p>
<pre>platform: vsphere: failureDomains:</pre>	<p>Establishes the relationships between a region and zone. You define a failure domain by using vCenter objects, such as a datastore object. A failure domain defines the vCenter location for OpenShift Container Platform cluster nodes.</p> <p>Value: An array of failure domain configuration objects.</p>
<pre>platform: vsphere: failureDomains: name:</pre>	<p>The name of the failure domain.</p> <p>Value: String</p>

Parameter	Description
<pre>platform: vsphere: failureDomains: region:</pre>	<p>If you define multiple failure domains for your cluster, you must attach the tag to each vCenter data center. To define a region, use a tag from the openshift-region tag category. For a single vSphere data center environment, you do not need to attach a tag, but you must enter an alphanumeric value, such as datacenter, for the parameter. If you want to base your failure domains on host groups, attach these tags to your vSphere clusters instead of your data centers.</p> <p>IMPORTANT</p> <p>OpenShift zones support for vSphere host groups is a Technology Preview feature only. Technology Preview features are not supported with Red Hat production service level agreements (SLAs) and might not be functionally complete. Red Hat does not recommend using them in production. These features provide early access to upcoming product features, enabling customers to test functionality and provide feedback during the development process.</p> <p>For more information about the support scope of Red Hat Technology Preview features, see Technology Preview Features Support Scope.</p> <p>Value: String</p>

Parameter	Description
<pre>platform: vsphere: failureDomains: regionType:</pre>	<p>Specifies the ComputeCluster region type to enable host groups.</p> <p>IMPORTANT</p> <p>OpenShift zones support for vSphere host groups is a Technology Preview feature only. Technology Preview features are not supported with Red Hat production service level agreements (SLAs) and might not be functionally complete. Red Hat does not recommend using them in production. These features provide early access to upcoming product features, enabling customers to test functionality and provide feedback during the development process.</p> <p>For more information about the support scope of Red Hat Technology Preview features, see Technology Preview Features Support Scope.</p>
<pre>platform: vsphere: failureDomains: server:</pre>	<p>Value: String</p> <p>Specifies the fully-qualified hostname or IP address of the VMware vCenter server, so that a client can access failure domain resources. You must apply the server role to the vSphere vCenter server location.</p> <p>Value: String</p>

Parameter	Description
platform: vsphere: failureDomains: zone:	<p>If you define multiple failure domains for your cluster, you must attach a tag to each vCenter cluster. To define a zone, use a tag from the openshift-zone tag category. For a single vSphere data center environment, you do not need to attach a tag, but you must enter an alphanumeric value, such as cluster, for the parameter. If you want to base your failure domains on host groups, define zones that correspond to your host groups instead of your clusters. Use these tags to associate each ESXi host with its host group.</p> <p>IMPORTANT</p> <p>OpenShift zones support for vSphere host groups is a Technology Preview feature only. Technology Preview features are not supported with Red Hat production service level agreements (SLAs) and might not be functionally complete. Red Hat does not recommend using them in production. These features provide early access to upcoming product features, enabling customers to test functionality and provide feedback during the development process.</p> <p>For more information about the support scope of Red Hat Technology Preview features, see Technology Preview Features Support Scope.</p> <p>Value: String</p>

Parameter	Description
<pre>platform: vsphere: failureDomains: zoneType:</pre>	<p>Specifies the HostGroup zone type to enable host groups.</p> <p>IMPORTANT</p> <p>OpenShift zones support for vSphere host groups is a Technology Preview feature only. Technology Preview features are not supported with Red Hat production service level agreements (SLAs) and might not be functionally complete. Red Hat does not recommend using them in production. These features provide early access to upcoming product features, enabling customers to test functionality and provide feedback during the development process.</p> <p>For more information about the support scope of Red Hat Technology Preview features, see Technology Preview Features Support Scope.</p>
	<p>Value: String</p> <p>The path to the vSphere compute cluster.</p> <p>Value: String</p>
<pre>platform: vsphere: failureDomains: topology: computeCluster:</pre>	<p>Lists and defines the data centers where OpenShift Container Platform virtual machines (VMs) operate. The list of data centers must match the list of data centers specified in the vcenters field.</p> <p>Value: String</p>
<pre>platform: vsphere: failureDomains: topology: datastore:</pre>	<p>Specifies the path to a vSphere datastore that stores virtual machines files for a failure domain. You must apply the datastore role to the vSphere vCenter datastore location.</p> <p>Value: String</p>

Parameter	Description
<pre>platform: vsphere: failureDomains: topology: folder:</pre>	<p>Optional: The absolute path of an existing folder where the user creates the virtual machines, for example, <code>/<data_center_name>/vm/<folder_name>/<subfolder_name></code>. If you do not provide this value, the installation program creates a top-level folder in the data center virtual machine folder that is named with the infrastructure ID. If you are providing the infrastructure for the cluster and you do not want to use the default StorageClass object, named thin, you can omit the folder parameter from the install-config.yaml file.</p> <p>Value: String</p>
<pre>platform: vsphere: failureDomains: topology: hostGroup:</pre>	<p>Specifies the vSphere host group to associate with the failure domain.</p> <p>IMPORTANT</p> <p>OpenShift zones support for vSphere host groups is a Technology Preview feature only. Technology Preview features are not supported with Red Hat production service level agreements (SLAs) and might not be functionally complete. Red Hat does not recommend using them in production. These features provide early access to upcoming product features, enabling customers to test functionality and provide feedback during the development process.</p> <p>For more information about the support scope of Red Hat Technology Preview features, see Technology Preview Features Support Scope.</p> <p>Value: String</p>
<pre>platform: vsphere: failureDomains: topology: networks:</pre>	<p>Lists any network in the vCenter instance that contains the virtual IP addresses and DNS records that you configured.</p> <p>Value: String</p>

Parameter	Description
<pre>platform: vsphere: failureDomains: topology: resourcePool:</pre>	<p>Optional: The absolute path of an existing resource pool where the installation program creates the virtual machines, for example, /<data_center_name>/host/<cluster_name>/Resources/<resource_pool_name>/<optional_nested_resource_pool_name>. If you do not specify a value, the installation program installs the resources in the root of the cluster under /<data_center_name>/host/<cluster_name>/Resources.</p> <p>Value: String</p>
<pre>platform: vsphere: failureDomains: topology template:</pre>	<p>Specifies the absolute path to a pre-existing Red Hat Enterprise Linux CoreOS (RHCOS) image template or virtual machine. The installation program can use the image template or virtual machine to quickly install RHCOS on vSphere hosts. Consider using this parameter as an alternative to uploading an RHCOS image on vSphere hosts. This parameter is available for use only on installer-provisioned infrastructure.</p> <p>Value: String</p>
<pre>platform: vsphere: ingressVIPs:</pre>	<p>Virtual IP (VIP) addresses that you configured for cluster Ingress.</p> <p> NOTE</p> <p>This parameter applies only to installer-provisioned infrastructure without an external load balancer configured. You must not specify this parameter in user-provisioned infrastructure.</p> <p>Value: Multiple IP addresses</p>
<pre>platform: vsphere: vcenters:</pre>	<p>Configures the connection details so that services can communicate with a vCenter server.</p> <p>Value: An array of vCenter configuration objects.</p>

Parameter	Description
<pre>platform: vsphere: vcenters: datacenters:</pre>	<p>Lists and defines the data centers where OpenShift Container Platform virtual machines (VMs) operate. The list of data centers must match the list of data centers specified in the failureDomains field.</p> <p>Value: String</p>
<pre>platform: vsphere: vcenters: password:</pre>	<p>The password associated with the vSphere user.</p> <p>Value: String</p>
<pre>platform: vsphere: vcenters: port:</pre>	<p>The port number used to communicate with the vCenter server.</p> <p>Value: Integer</p>
<pre>platform: vsphere: vcenters: server:</pre>	<p>The fully qualified host name (FQHN) or IP address of the vCenter server.</p> <p>Value: String</p>
<pre>platform: vsphere: vcenters: user:</pre>	<p>The username associated with the vSphere user.</p> <p>Value: String</p>

9.1.5. Deprecated VMware vSphere configuration parameters

In OpenShift Container Platform 4.13, the following vSphere configuration parameters are deprecated. You can continue to use these parameters, but the installation program does not automatically specify these parameters in the **install-config.yaml** file.

The following table lists each deprecated vSphere configuration parameter:

Table 9.5. Deprecated VMware vSphere cluster parameters

Parameter	Description
<pre>platform: vsphere: apiVIP:</pre>	<p>The virtual IP (VIP) address that you configured for control plane API access.</p> <p></p> <p>NOTE</p> <p>In OpenShift Container Platform 4.12 and later, the apiVIP configuration setting is deprecated. Instead, use a List format to enter a value in the apiVIPs configuration setting.</p> <p>Value: An IP address, for example 128.0.0.1.</p>
<pre>platform: vsphere: cluster:</pre>	<p>The vCenter cluster to install the OpenShift Container Platform cluster in.</p> <p>Value: String</p>
<pre>platform: vsphere: datacenter:</pre>	<p>Defines the data center where OpenShift Container Platform virtual machines (VMs) operate.</p> <p>Value: String</p>
<pre>platform: vsphere: defaultDatastore:</pre>	<p>The name of the default datastore to use for provisioning volumes.</p> <p>Value: String</p>
<pre>platform: vsphere: folder:</pre>	<p>Optional: The absolute path of an existing folder where the installation program creates the virtual machines. If you do not provide this value, the installation program creates a folder that is named with the infrastructure ID in the data center virtual machine folder.</p> <p>Value: String, for example, /<data_center_name>/vm/<folder_name>/<subfolder_name>.</p>

Parameter	Description
<pre>platform: vsphere: ingressVIP:</pre>	<p>Virtual IP (VIP) addresses that you configured for cluster Ingress.</p> <p> NOTE</p> <p>In OpenShift Container Platform 4.12 and later, the ingressVIP configuration setting is deprecated. Instead, use a List format to enter a value in the ingressVIPs configuration setting.</p> <p>Value: An IP address, for example 128.0.0.1.</p>
<pre>platform: vsphere: network:</pre>	<p>The network in the vCenter instance that contains the virtual IP addresses and DNS records that you configured.</p> <p>Value: String</p>
<pre>platform: vsphere: password:</pre>	<p>The password for the vCenter user name.</p> <p>Value: String</p>
<pre>platform: vsphere: resourcePool:</pre>	<p>Optional: The absolute path of an existing resource pool where the installation program creates the virtual machines. If you do not specify a value, the installation program installs the resources in the root of the cluster under /<data_center_name>/host/<cluster_name>/Resources.</p> <p>Value: String, for example, /<data_center_name>/host/<cluster_name>/Resources/<resource_pool_name>/<optional_nested_resource_pool_name>.</p>
<pre>platform: vsphere: username:</pre>	<p>The user name to use to connect to the vCenter instance with. This user must have at least the roles and privileges that are required for static or dynamic persistent volume provisioning in vSphere.</p> <p>Value: String</p>

Parameter	Description
platform: vsphere: vCenter:	The fully-qualified hostname or IP address of a vCenter server. Value: String

9.1.6. Optional VMware vSphere machine pool configuration parameters

Optional VMware vSphere machine pool configuration parameters are described in the following table:

Table 9.6. Optional VMware vSphere machine pool parameters

Parameter	Description
platform: vsphere: clusterOSImage:	The location from which the installation program downloads the Red Hat Enterprise Linux CoreOS (RHCOS) image. Before setting a path value for this parameter, ensure that the default RHCOS boot image in the OpenShift Container Platform release matches the RHCOS image template or virtual machine version; otherwise, cluster installation might fail. Value: An HTTP or HTTPS URL, optionally with a SHA-256 checksum. For example, <a href="https://mirror.openshift.com/images/rhcos-<version>-vmware.<architecture>.ova">https://mirror.openshift.com/images/rhcos-<version>-vmware.<architecture>.ova .
platform: vsphere: osDisk: diskSizeGB:	The size of the disk in gigabytes. Value: Integer
platform: vsphere: cpus:	The total number of virtual processor cores to assign a virtual machine. The value of platform.vsphere.cpus must be a multiple of platform.vsphere.coresPerSocket value. Value: Integer
platform: vsphere: coresPerSocket:	The number of cores per socket in a virtual machine, where platform.vsphere.cpus divided by platform.vsphere.coresPerSocket determines the number of virtual sockets on a virtual machine. Control plane nodes and compute nodes default to 4 virtual sockets on a virtual machine. Value: Integer

Parameter	Description
<pre>platform: vsphere: memoryMB:</pre>	<p>The size of a virtual machine's memory in megabytes.</p> <p>Value: Integer</p>
<pre>platform: vsphere: dataDisks: name:</pre>	<p>The name of the data disk to add to the virtual machines. The maximum name length is 80 characters.</p> <p>IMPORTANT</p> <p>Installing OpenShift Container Platform on VMware vSphere using multiple data disks is a Technology Preview feature only. Technology Preview features are not supported with Red Hat production service level agreements (SLAs) and might not be functionally complete. Red Hat does not recommend using them in production. These features provide early access to upcoming product features, enabling customers to test functionality and provide feedback during the development process.</p> <p>For more information about the support scope of Red Hat Technology Preview features, see Technology Preview Features Support Scope.</p>
	<p>Value: String</p>
<pre>platform: vsphere: dataDisks: sizeGiB:</pre>	<p>The size of the data disk to add to the virtual machines. The maximum size is 16384 GiB.</p> <p>Value: Integer</p>
<pre>platform: vsphere: dataDisks: provisioningMode:</pre>	<p>Optional: The data disk provisioning method. This value defaults to the vSphere default storage policy, if not set.</p> <p>Value: Valid values are Thin, Thick, or EagerlyZeroed.</p>

CHAPTER 10. MULTIPLE REGIONS AND ZONES CONFIGURATION FOR A CLUSTER ON VMWARE VSPHERE

As an administrator, you can specify multiple regions and zones for your OpenShift Container Platform cluster that runs on a VMware vSphere instance. This configuration reduces the risk of a hardware failure or network outage causing your cluster to fail.

A failure domain configuration lists parameters that create a topology. The following list states some of these parameters:

- **computeCluster**
- **datacenter**
- **datastore**
- **networks**
- **resourcePool**

After you define multiple regions and zones for your OpenShift Container Platform cluster, you can create or migrate nodes to another failure domain.



IMPORTANT

If you want to migrate pre-existing OpenShift Container Platform cluster compute nodes to a failure domain, you must define a new compute machine set for the compute node. This new machine set can scale up a compute node according to the topology of the failure domain, and scale down the pre-existing compute node.

The cloud provider adds **topology.kubernetes.io/zone** and **topology.kubernetes.io/region** labels to any compute node provisioned by a machine set resource.

For more information, see [Creating a compute machine set](#).

10.1. SPECIFYING MULTIPLE REGIONS AND ZONES FOR YOUR CLUSTER ON VSPHERE

You can configure the **infrastructures.config.openshift.io** configuration resource to specify multiple regions and zones for your OpenShift Container Platform cluster that runs on a VMware vSphere instance.

Topology-aware features for the cloud controller manager and the vSphere Container Storage Interface (CSI) Operator Driver require information about the vSphere topology where you host your OpenShift Container Platform cluster. This topology information exists in the **infrastructures.config.openshift.io** configuration resource.

Before you specify regions and zones for your cluster, you must ensure that all data centers and compute clusters contain tags, so that the cloud provider can add labels to your node. For example, if **data-center-1** represents **region-a** and **compute-cluster-1** represents **zone-1**, the cloud provider adds an **openshift-region** category label with a value of **region-a** to **data-center-1**. Additionally, the cloud provider adds an **openshift-zone** category tag with a value of **zone-1** to **compute-cluster-1**.



NOTE

You can migrate control plane nodes with vMotion capabilities to a failure domain. After you add these nodes to a failure domain, the cloud provider adds **topology.kubernetes.io/zone** and **topology.kubernetes.io/region** labels to these nodes.

Prerequisites

- You created the **openshift-region** and **openshift-zone** tag categories on the vCenter server.
- You ensured that each data center and compute cluster contains tags that represent the name of their associated region or zone, or both.
- Optional: If you defined **API** and **Ingress** static IP addresses to the installation program, you must ensure that all regions and zones share a common layer 2 network. This configuration ensures that API and Ingress Virtual IP (VIP) addresses can interact with your cluster.



IMPORTANT

If you do not supply tags to all data centers and compute clusters before you create a node or migrate a node, the cloud provider cannot add the **topology.kubernetes.io/zone** and **topology.kubernetes.io/region** labels to the node. This means that services cannot route traffic to your node.

Procedure

1. Edit the **infrastructures.config.openshift.io** custom resource definition (CRD) of your cluster to specify multiple regions and zones in the **failureDomains** section of the resource by running the following command:

```
$ oc edit infrastructures.config.openshift.io cluster
```

Example infrastructures.config.openshift.io CRD for a instance named `cluster` with multiple regions and zones defined in its configuration

```
spec:
  cloudConfig:
    key: config
    name: cloud-provider-config
  platformSpec:
    type: vSphere
  vsphere:
    vcenters:
      - datacenters:
          - <region_a_data_center>
          - <region_b_data_center>
        port: 443
        server: <your_vcenter_server>
  failureDomains:
    - name: <failure_domain_1>
      region: <region_a>
      zone: <zone_a>
      server: <your_vcenter_server>
```

```

topology:
  datacenter: <region_a_dc>
  computeCluster: "</region_a_dc/host/zone_a_cluster>"
  resourcePool: "</region_a_dc/host/zone_a_cluster/Resources/resource_pool>"
  datastore: "</region_a_dc/datastore/datastore_a>"
  networks:
    - port-group
  - name: <failure_domain_2>
    region: <region_a>
    zone: <zone_b>
    server: <your_vcenter_server>
    topology:
      computeCluster: </region_a_dc/host/zone_b_cluster>
      datacenter: <region_a_dc>
      datastore: </region_a_dc/datastore/datastore_a>
      networks:
        - port-group
      - name: <failure_domain_3>
        region: <region_b>
        zone: <zone_a>
        server: <your_vcenter_server>
        topology:
          computeCluster: </region_b_dc/host/zone_a_cluster>
          datacenter: <region_b_dc>
          datastore: </region_b_dc/datastore/datastore_b>
          networks:
            - port-group
    nodeNetworking:
      external: {}
      internal: {}

```



IMPORTANT

After you create a failure domain and you define it in a CRD for a VMware vSphere cluster, you must not modify or delete the failure domain. Doing any of these actions with this configuration can impact the availability and fault tolerance of a control plane machine.

2. Save the resource file to apply the changes.

Additional resources

- [Parameters for the cluster-wide infrastructure CRD](#)

10.2. ENABLING A MULTIPLE LAYER 2 NETWORK FOR YOUR CLUSTER

You can configure your cluster to use a multiple layer 2 network configuration so that data transfer among nodes can span across multiple networks.

Prerequisites

- You configured network connectivity among machines so that cluster components can communicate with each other.

Procedure

- If you installed your cluster with installer-provisioned infrastructure, you must ensure that all control plane nodes share a common layer 2 network. Additionally, ensure compute nodes that are configured for Ingress pod scheduling share a common layer 2 network.
 - If you need compute nodes to span multiple layer 2 networks, you can create infrastructure nodes that can host Ingress pods.
 - If you need to provision workloads across additional layer 2 networks, you can create compute machine sets on vSphere and then move these workloads to your target layer 2 networks.
- If you installed your cluster on infrastructure that you provided, which is defined as a user-provisioned infrastructure, complete the following actions to meet your needs:
 - Configure your API load balancer and network so that the load balancer can reach the API and Machine Config Server on the control plane nodes.
 - Configure your Ingress load balancer and network so that the load balancer can reach the Ingress pods on the compute or infrastructure nodes.

Additional resources

- [Installing a cluster on vSphere with network customizations](#)
- [Creating a compute machine set](#)

10.3. PARAMETERS FOR THE CLUSTER-WIDE INFRASTRUCTURE CRD

You must set values for specific parameters in the cluster-wide infrastructure, **infrastructures.config.openshift.io**, Custom Resource Definition (CRD) to define multiple regions and zones for your OpenShift Container Platform cluster that runs on a VMware vSphere instance.

The following table lists mandatory parameters for defining multiple regions and zones for your OpenShift Container Platform cluster:

Parameter	Description
vcenters	The vCenter servers for your OpenShift Container Platform cluster. You can specify either a single vCenter, or up to 3 vCenters.
datacenters	vCenter data centers where VMs associated with the OpenShift Container Platform cluster will be created or presently exist.
port	The TCP port of the vCenter server.
server	The fully qualified domain name (FQDN) of the vCenter server.
failureDomains	The list of failure domains.
name	The name of the failure domain.

Parameter	Description
region	The value of the openshift-region tag assigned to the topology for the failure failure domain.
zone	The value of the openshift-zone tag assigned to the topology for the failure failure domain.
topology	The vCenter resources associated with the failure domain.
datacenter	The data center associated with the failure domain.
computeCluster	The full path of the compute cluster associated with the failure domain.
resourcePool	The full path of the resource pool associated with the failure domain.
datastore	The full path of the datastore associated with the failure domain.
networks	A list of port groups associated with the failure domain. Only one portgroup may be defined.

Additional resources

- [Specifying multiple regions and zones for your cluster on vSphere](#)

10.4. SPECIFYING MULTIPLE HOST GROUPS FOR YOUR CLUSTER ON VSphere

You can configure the **infrastructures.config.openshift.io** configuration resource to specify multiple host groups for your OpenShift Container Platform cluster that runs on a VMware vSphere instance. This is necessary if your vSphere instance is in a stretched cluster configuration, with your ESXi hosts and storage distributed across multiple physical data centers. Use this procedure if you did not already configure host groups for your OpenShift Container Platform cluster at installation, or if you need to update your OpenShift Container Platform cluster with additional host groups.



IMPORTANT

OpenShift zones support for vSphere host groups is a Technology Preview feature only. Technology Preview features are not supported with Red Hat production service level agreements (SLAs) and might not be functionally complete. Red Hat does not recommend using them in production. These features provide early access to upcoming product features, enabling customers to test functionality and provide feedback during the development process.

For more information about the support scope of Red Hat Technology Preview features, see [Technology Preview Features Support Scope](#).

Prerequisites

- ESXi hosts are grouped into host groups, which are linked via VM-host affinity rules to corresponding virtual machine (VM) groups. See the following example **govc** commands for details:

```
# This example shows the correct configuration for a cluster with two host groups:

# Create host groups:
govc cluster.group.create -name <host_group_1> -host
govc cluster.group.create -name <host_group_2> -host

# Create VM groups:
govc cluster.group.create -name <vm_group_1> -vm
govc cluster.group.create -name <vm_group_2> -vm

# Create VM-host affinity rules:
govc cluster.rule.create -name <rule_1> -enable -vm-host -vm-group <vm_group_1> -host-affine-group <host_group_1>
govc cluster.rule.create -name <rule_2> -enable -vm-host -vm-group <vm_group_2> -host-affine-group <host_group_2>

# Add ESXi hosts to host groups:
govc cluster.group.change -name <host_group_1> <esxi_host_1_ip>
govc cluster.group.change -name <host_group_2> <esxi_host_2_ip>
```

- openshift-region** and **openshift-zone** tag categories are created on the vCenter server.
- Compute clusters have tags from the **openshift-region** tag category.
- ESXi hosts within host groups have tags from the **openshift-zone** tag category.
- Host.Inventory.EditCluster** privilege is granted on the vSphere vCenter cluster object.
- TechPreviewNoUpgrade** feature set is enabled. For more information, "see Enabling features using feature gates".

Procedure

- Edit the infrastructure settings of your OpenShift Container Platform cluster.

- To copy your existing infrastructure settings to a file, run the following command:

```
$ oc get infrastructures.config.openshift.io cluster -o yaml >
<name_of_infrastructure_file>.yaml
```

- Edit your infrastructure file to include a failure domain for each host group in your vSphere cluster. Refer to the following YAML file for an example of this configuration. Ensure you replace any values wrapped in angle brackets (< >) with your values:

```
apiVersion: config.openshift.io/v1
kind: Infrastructure
metadata:
  name: cluster
spec:
  cloudConfig:
    key: config
```

```

name: cloud-provider-config
platformSpec:
  type: VSphere
  vsphere:
    apiServerInternalIPs:
      - <internal_ip_of_api_server>
    failureDomains:
      - name: <unique_name_for_failure_domain_1>
        region: <cluster_1_region_tag>
        server: <vcenter_server_ip_address>
    zoneAffinity:
      type: HostGroup
      hostGroup:
        vmGroup: <name_of_vm_group_1>
        hostGroup: <name_of_host_group_1>
        vmHostRule: <name_of_vm_host_affinity_rule_1>
    regionAffinity:
      type: ComputeCluster
    topology:
      computeCluster: /<data_center_1>/host/<cluster_1>
      datacenter: <data_center_1>
      datastore: /<data_center_1>/datastore/<datastore_1>
      networks:
        - VM Network
        resourcePool: /<data_center_1>/host/<cluster_1>/Resources
        template: /<data_center_1>/vm/<vm_template>
      zone: <host_group_1_tag>
    - name: <unique_name_for_failure_domain_2>
      region: <cluster_1_region_tag>
      server: <vcenter_server_ip_address>
    zoneAffinity:
      type: HostGroup
      hostGroup:
        vmGroup: <name_of_vm_group_2>
        hostGroup: <name_of_host_group_2>
        vmHostRule: <name_of_vm_host_affinity_rule_2>
    regionAffinity:
      type: ComputeCluster
    topology:
      computeCluster: /<data_center_1>/host/<cluster_1>
      datacenter: <data_center_1>
      datastore: /<data_center_1>/datastore/<datastore_1>
      networks:
        - VM Network
        resourcePool: /<data_center_1>/host/<cluster_1>/Resources
        template: /<data_center_1>/vm/<vm_template>
      zone: <host_group_2_tag>
# ...

```

c. To update your cluster with these changes, run the following command:

```
$ oc replace -f <name_of_infrastructure_file>.yaml
```

2. Update your **ControlPlaneMachineSet** custom resource (CR) with the new failure domains by completing the following steps:

- Edit the **ControlPlaneMachineSet** CR by running the following command:

```
$ oc edit controlplanemachinesets.machine.openshift.io -n openshift-machine-api cluster
```

- Edit the **failureDomains** parameter as shown in the following example:

```
spec:
  replicas: 3
  selector:
    matchLabels:
      machine.openshift.io/cluster-api-cluster: jdoe3-whb8l
      machine.openshift.io/cluster-api-machine-role: master
      machine.openshift.io/cluster-api-machine-type: master
  state: Active
  strategy:
    type: RollingUpdate
  template:
    machineType: machines_v1beta1_machine.openshift.io
    machines_v1beta1_machine.openshift.io:
      failureDomains:
        platform: VSphere
        vsphere:
          - name: <failure_domain_1_name>
          - name: <failure_domain_2_name>
# ...
```

- Verify that your control plane nodes have finished updating before proceeding further. To do this, run the following command:

```
$ oc get controlplanemachinesets.machine.openshift.io -n openshift-machine-api
```

- Create new **MachineSet** CRs for your failure domains.

- To retrieve the configuration of an existing **MachineSet** CR for use as a template, run the following command:

```
$ oc get machinesets.machine.openshift.io -n openshift-machine-api
<existing_machine_set> -o yaml > machineset-<failure_domain_name>.yaml
```

- Copy the template as needed to create **MachineSet** CR files for each failure domain that you defined in your infrastructure file. Refer to the following example:

```
apiVersion: machine.openshift.io/v1beta1
kind: MachineSet
metadata:
  labels:
    machine.openshift.io/cluster-api-cluster: <infrastructure_id>
  name: <machineset_name>
  namespace: openshift-machine-api
spec:
  replicas: 0
  selector:
    matchLabels:
      machine.openshift.io/cluster-api-cluster: <infrastructure_id>
```

```

machine.openshift.io/cluster-api-machineset: <machineset_name>
template:
metadata:
labels:
  machine.openshift.io/cluster-api-cluster: <infrastructure_id>
  machine.openshift.io/cluster-api-machine-role: worker
  machine.openshift.io/cluster-api-machine-type: worker
  machine.openshift.io/cluster-api-machineset: <machineset_name>
spec:
lifecycleHooks: {}
metadata: {}
providerSpec:
value:
  apiVersion: machine.openshift.io/v1beta1
  credentialsSecret:
    name: vsphere-cloud-credentials
  diskGiB: <disk_GiB>
  kind: VSphereMachineProviderSpec
  memoryMiB: <memory_in_MiB>
  metadata:
    creationTimestamp: null
  network:
    devices:
      - networkName: VM Network
  numCPUs: <number_of_cpus>
  numCoresPerSocket: <number_of_cores_per_socket>
  snapshot: ""
  template: <template_name>
  userDataSecret:
    name: worker-user-data
  workspace:
    datacenter: <data_center_1>
    datastore: /<data_center_1>/datastore/<datastore_1>
    folder: /<data_center_1>/vm/<folder>
    resourcePool: /<data_center_1>/host/<cluster_1>/Resources
    server: <server_ip_address>
    vmGroup: <name_of_vm_group_1>
# ...

```

c. For each **MachineSet** CR file, run the following command:

```
$ oc create -f <name_of_machine_set_file>.yaml
```

Additional resources

- [Enabling features using feature gates](#)

CHAPTER 11. ENABLING ENCRYPTION ON A VSphere CLUSTER

You can encrypt your virtual machines after installing OpenShift Container Platform 4.19 on vSphere by draining and shutting down your nodes one at a time. While each virtual machine is shutdown, you can enable encryption in the vCenter web interface.

11.1. ENCRYPTING VIRTUAL MACHINES

You can encrypt your virtual machines with the following process. You can drain your virtual machines, power them down and encrypt them using the vCenter interface. Finally, you can create a storage class to use the encrypted storage.

Prerequisites

- You have configured a Standard key provider in vSphere. For more information, see [Adding a KMS to vCenter Server](#).



IMPORTANT

The Native key provider in vCenter is not supported. For more information, see [vSphere Native Key Provider Overview](#).

- You have enabled host encryption mode on all of the ESXi hosts that are hosting the cluster. For more information, see [Enabling host encryption mode](#).
- You have a vSphere account which has all cryptographic privileges enabled. For more information, see [Cryptographic Operations Privileges](#).

Procedure

1. Drain and cordon one of your nodes. For detailed instructions on node management, see "Working with Nodes".
2. Shutdown the virtual machine associated with that node in the vCenter interface.
3. Right-click on the virtual machine in the vCenter interface and select **VM Policies** → **Edit VM Storage Policies**.
4. Select an encrypted storage policy and select **OK**.
5. Start the encrypted virtual machine in the vCenter interface.
6. Repeat steps 1-5 for all nodes that you want to encrypt.
7. Configure a storage class that uses the encrypted storage policy. For more information about configuring an encrypted storage class, see "VMware vSphere CSI Driver Operator".

11.2. ADDITIONAL RESOURCES

- [Working with nodes](#)
- [vSphere encryption](#)

- Requirements for encrypting virtual machines

CHAPTER 12. CONFIGURING THE VSphere CONNECTION SETTINGS AFTER AN INSTALLATION

After installing an OpenShift Container Platform cluster on vSphere with the platform integration feature enabled, you might need to update the vSphere connection settings manually, depending on the installation method.

For installations using the Assisted Installer, you must update the connection settings. This is because the Assisted Installer adds default connection settings to the **vSphere connection configuration** wizard as placeholders during the installation.

For installer-provisioned or user-provisioned infrastructure installations, you should have entered valid connection settings during the installation. You can use the **vSphere connection configuration** wizard at any time to validate or modify the connection settings, but this is not mandatory for completing the installation.

12.1. CONFIGURING THE VSphere CONNECTION SETTINGS

Modify the following vSphere configuration settings as required:

- vCenter address
- vCenter cluster
- vCenter username
- vCenter password
- vCenter address
- vSphere data center
- vSphere datastore
- Virtual machine folder

Prerequisites

- The Assisted Installer has finished installing the cluster successfully.
- The cluster is connected to <https://console.redhat.com>.

Procedure

1. In the Administrator perspective, navigate to **Home** → **Overview**.
2. Under **Status**, click **vSphere connection** to open the **vSphere connection configuration** wizard.
3. In the **vCenter** field, enter the network address of the vSphere vCenter server. This can be either a domain name or an IP address. It appears in the vSphere web client URL; for example [https://\[your_vCenter_address\]/ui](https://[your_vCenter_address]/ui).
4. In the **vCenter cluster** field, enter the name of the vSphere vCenter cluster where OpenShift Container Platform is installed.



IMPORTANT

This step is mandatory if you installed OpenShift Container Platform 4.13 or later.

5. In the **Username** field, enter your vSphere vCenter username.
6. In the **Password** field, enter your vSphere vCenter password.



WARNING

The system stores the username and password in the **vsphere-creds** secret in the **kube-system** namespace of the cluster. An incorrect vCenter username or password makes the cluster nodes unschedulable.

7. In the **Datacenter** field, enter the name of the vSphere data center that contains the virtual machines used to host the cluster; for example, **SDDC-Datacenter**.
8. In the **Default data store** field, enter the path and name of the vSphere data store that stores the persistent data volumes; for example, **/SDDC-Datacenter/datastore/datasstorename**.



WARNING

Updating the vSphere data center or default data store after the configuration has been saved detaches any active vSphere **PersistentVolumes**.

9. In the **Virtual Machine Folder** field, enter the data center folder that contains the virtual machine of the cluster; for example, **/SDDC-Datacenter/vm/ci-1n-hjg4vg2-c61657-t2gxr**. For the OpenShift Container Platform installation to succeed, all virtual machines comprising the cluster must be located in a single data center folder.
10. Click **Save Configuration**. This updates the **cloud-provider-config** ConfigMap resource in the **openshift-config** namespace, and starts the configuration process.
11. Reopen the **VSphere connection configuration** wizard and expand the **Monitored operators** panel. Check that the status of the operators is either **Progressing** or **Healthy**.

12.2. VERIFYING THE CONFIGURATION

The connection configuration process updates operator statuses and control plane nodes. It takes approximately an hour to complete. During the configuration process, the nodes will reboot. Previously bound **PersistentVolumeClaims** objects might become disconnected.

Prerequisites

- You have saved the configuration settings in the **vSphere connection configuration** wizard.

Procedure

1. Check that the configuration process completed successfully:
 - a. In the OpenShift Container Platform Administrator perspective, navigate to **Home** → **Overview**.
 - b. Under **Status** click **Operators**. Wait for all operator statuses to change from **Progressing** to **All succeeded**. A **Failed** status indicates that the configuration failed.
 - c. Under **Status**, click **Control Plane**. Wait for the response rate of all Control Plane components to return to 100%. A **Failed** control plane component indicates that the configuration failed.
2. A failure indicates that at least one of the connection settings is incorrect. Change the settings in the **vSphere connection configuration** wizard and save the configuration again.
2. Check that you are able to bind **PersistentVolumeClaims** objects by performing the following steps:

- a. Create a **StorageClass** object using the following YAML:

```
kind: StorageClass
apiVersion: storage.k8s.io/v1
metadata:
  name: vsphere-sc
provisioner: kubernetes.io/vsphere-volume
parameters:
  datastore: YOURVCENTERDATASTORE
  diskformat: thin
  reclaimPolicy: Delete
  volumeBindingMode: Immediate
```

- b. Create a **PersistentVolumeClaims** object using the following YAML:

```
kind: PersistentVolumeClaim
apiVersion: v1
metadata:
  name: test-pvc
  namespace: openshift-config
  annotations:
    volume.beta.kubernetes.io/storage-provisioner: kubernetes.io/vsphere-volume
  finalizers:
    - kubernetes.io/pvc-protection
spec:
  accessModes:
    - ReadWriteOnce
  resources:
    requests:
      storage: 10Gi
  storageClassName: vsphere-sc
  volumeMode: Filesystem
```

If you are unable to create a **PersistentVolumeClaims** object, you can troubleshoot by navigating to **Storage** → **PersistentVolumeClaims** in the **Administrator** perspective of the OpenShift Container Platform web console.

For instructions on creating storage objects, see [Dynamic provisioning](#).