OpenShift Container Platform 4.12

Installing on VMC

Installing OpenShift Container Platform on VMware Cloud

Last Updated: 2024-09-11

OpenShift Container Platform 4.12 Installing on VMC

Installing OpenShift Container Platform on VMware Cloud

Abstract

This document describes how to install OpenShift Container Platform on VMware Cloud.

Table of Contents

CHAPTER 1. PREPARING TO INSTALL ON VMC

1.1. PREREQUISITES

- 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.
- If you use a firewall and plan to use Telemetry, you configured the firewall to allow the sites required by your cluster.

1.2. CHOOSING A METHOD TO INSTALL OPENSHIFT CONTAINER PLATFORM ON VMC

You can install OpenShift Container Platform on VMC by using installer-provisioned or user-provisioned infrastructure. The default installation type uses installer-provisioned infrastructure, where the installation program provisions the underlying infrastructure for the cluster. You can also install OpenShift Container Platform on infrastructure that you provide. If you do not use infrastructure that the installation program provisions, you must manage and maintain the cluster resources yourself.

See the Installation process for more information about installer-provisioned and user-provisioned installation processes.

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 VMC 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.

1.2.1. Installer-provisioned infrastructure installation of OpenShift Container Platform on VMC

Installer-provisioned infrastructure allows the installation program to pre-configure and automate the provisioning of resources required by OpenShift Container Platform.

- **Installing a cluster on VMC**: You can install OpenShift Container Platform on VMC by using installer-provisioned infrastructure installation with no customization.
- **Installing a cluster on VMC with customizations**: You can install OpenShift Container Platform on VMC by using installer-provisioned infrastructure installation with the default customization options.
- **Installing a cluster on VMC with network customizations**: You can install OpenShift Container Platform on installer-provisioned VMC 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 VMC in a restricted network**: You can install a cluster on VMC 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.

1.2.2. User-provisioned infrastructure installation of OpenShift Container Platform on VMC

User-provisioned infrastructure requires the user to provision all resources required by OpenShift Container Platform.

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

1.3. 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 7.0 Update 2 or later
- Version 8.0 Update 1 or later

You can host the VMware vSphere infrastructure on-premise or on a VMware Cloud Verified provider that meets the requirements outlined in the following table:

Table 1.1. Version requirements for vSphere virtual environments

Virtual environment product	Required version
VMware virtual hardware	15 or later
vSphere ESXi hosts	7.0 Update 2 or later; 8.0 Update 1 or later
vCenter host	7.0 Update 2 or later; 8.0 Update 1 or later

IMPORTANT

Installing a cluster on VMware vSphere versions 7.0 and 7.0 Update 1 is deprecated. These versions are still fully supported, but all vSphere 6.x versions are no longer supported. Version 4.12 of OpenShift Container Platform requires VMware virtual hardware version 15 or later. To update the hardware version for your vSphere virtual machines, see the "Updating hardware on nodes running in vSphere" article in the *Updating clusters* section.

Table 1.2. Minimum supported vSphere version for VMware components

Component	Minimum supported versions	Description
Hypervisor	vSphere 7.0 Update 2 (or later) or vSphere 8.0 Update 1 (or later) with virtual hardware version 15	This 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.
Storage with in-tree drivers	vSphere 7.0 Update 2 or later; 8.0 Update 1 or later	This plugin creates vSphere storage by using the in-tree storage drivers for vSphere included in OpenShift Container Platform.

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.

1.4. VMWARE VSPHERE CSI DRIVER OPERATOR REQUIREMENTS

To install the vSphere CSI Driver Operator, the following requirements must be met:

- VMware vSphere version: 7.0 Update 2 or later; 8.0 Update 1 or later
- vCenter version: 7.0 Update 2 or later; 8.0 Update 1 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.

Additional resources

- To remove a third-party CSI driver, see Removing a third-party vSphere CSI Driver .

1.5. UNINSTALLING AN INSTALLER-PROVISIONED INFRASTRUCTURE INSTALLATION OF OPENSHIFT CONTAINER PLATFORM ON VMC

- Uninstalling a cluster on VMC that uses installer-provisioned infrastructure You can remove a cluster that you deployed on VMC infrastructure that used installer-provisioned infrastructure.

CHAPTER 2. INSTALLING A CLUSTER ON VMC

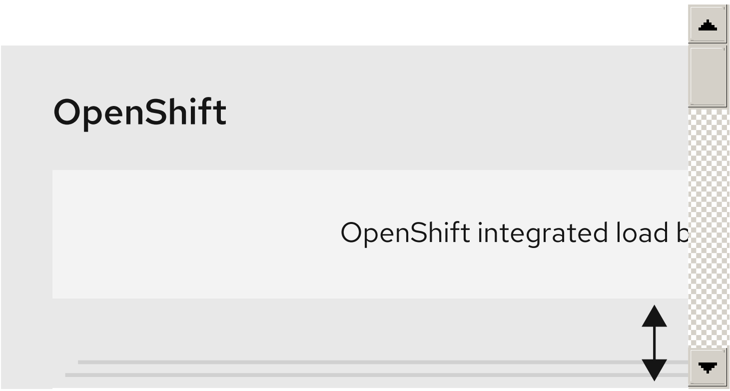
In OpenShift Container Platform version 4.12, you can install a cluster on VMware vSphere by deploying it to VMware Cloud (VMC) on AWS.

NOTE

OpenShift Container Platform supports deploying a cluster to a single VMware vCenter only. Deploying a cluster with machines/machine sets on multiple vCenters is not supported.

2.1. SETTING UP VMC FOR VSPHERE

You can install OpenShift Container Platform on VMware Cloud (VMC) on AWS hosted vSphere clusters to enable applications to be deployed and managed both on-premise and off-premise, across the hybrid cloud.



You must configure several options in your VMC environment prior to installing OpenShift Container Platform on VMware vSphere. Ensure your VMC environment has the following prerequisites:

- Create a non-exclusive, DHCP-enabled, NSX-T network segment and subnet. Other virtual machines (VMs) can be hosted on the subnet, but at least eight IP addresses must be available for the OpenShift Container Platform deployment.
- Allocate two IP addresses, outside the DHCP range, and configure them with reverse DNS records.
  - A DNS record for **api.<cluster_name>.<base_domain>** pointing to the allocated IP address.
  - A DNS record for ***.apps.<cluster_name>.<base_domain>** pointing to the allocated IP address.
- Configure the following firewall rules:
  - An ANY:ANY firewall rule between the OpenShift Container Platform compute network and the internet. This is used by nodes and applications to download container images.
  - An ANY:ANY firewall rule between the installation host and the software-defined data center (SDDC) management network on port 443. This allows you to upload the Red Hat Enterprise Linux CoreOS (RHCOS) OVA during deployment.
  - An HTTPS firewall rule between the OpenShift Container Platform compute network and vCenter. This connection allows OpenShift Container Platform to communicate with

vCenter for provisioning and managing nodes, persistent volume claims (PVCs), and other resources.

- You must have the following information to deploy OpenShift Container Platform:
  - The OpenShift Container Platform cluster name, such as **vmc-prod-1**.
  - The base DNS name, such as **companyname.com**.
  - If not using the default, the pod network CIDR and services network CIDR must be identified, which are set by default to **10.128.0.0/14** and **172.30.0.0/16**, respectively. These CIDRs are used for pod-to-pod and pod-to-service communication and are not accessible externally; however, they must not overlap with existing subnets in your organization.
  - The following vCenter information:
    - vCenter hostname, username, and password
    - Datacenter name, such as **SDDC-Datacenter**
    - Cluster name, such as **Cluster-1**
    - Network name
    - Datastore name, such as **WorkloadDatastore**

      NOTE

      It is recommended to move your vSphere cluster to the VMC **Compute-ResourcePool** resource pool after your cluster installation is finished.

- A Linux-based host deployed to VMC as a bastion.
  - The bastion host can be Red Hat Enterprise Linux (RHEL) or any another Linux-based host; it must have internet connectivity and the ability to upload an OVA to the ESXi hosts.
  - Download and install the OpenShift CLI tools to the bastion host.
    - The **openshift-install** installation program
    - The OpenShift CLI (**oc**) tool

NOTE

You cannot use the VMware NSX Container Plugin for Kubernetes (NCP), and NSX is not used as the OpenShift SDN. The version of NSX currently available with VMC is incompatible with the version of NCP certified with OpenShift Container Platform.

However, the NSX DHCP service is used for virtual machine IP management with the full-stack automated OpenShift Container Platform deployment and with nodes provisioned, either manually or automatically, by the Machine API integration with vSphere. Additionally, NSX firewall rules are created to enable access with the OpenShift Container Platform cluster and between the bastion host and the VMC vSphere hosts.

2.1.1. VMC Sizer tool

VMware Cloud on AWS is built on top of AWS bare metal infrastructure; this is the same bare metal infrastructure which runs AWS native services. When a VMware cloud on AWS software-defined data center (SDDC) is deployed, you consume these physical server nodes and run the VMware ESXi hypervisor in a single tenant fashion. This means the physical infrastructure is not accessible to anyone else using VMC. It is important to consider how many physical hosts you will need to host your virtual infrastructure.

To determine this, VMware provides the VMC on AWS Sizer. With this tool, you can define the resources you intend to host on VMC:

- Types of workloads
- Total number of virtual machines
- Specification information such as:
  - Storage requirements
  - vCPUs
  - vRAM
  - Overcommit ratios

With these details, the sizer tool can generate a report, based on VMware best practices, and recommend your cluster configuration and the number of hosts you will need.

2.2. VSPHERE PREREQUISITES

- 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 block registry storage. For more information on persistent storage, see Understanding persistent storage.
- 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. INTERNET ACCESS FOR OPENSHIFT CONTAINER PLATFORM

In OpenShift Container Platform 4.12, you require access to the internet to install your cluster.

You must have internet access to:

- Access OpenShift Cluster Manager Hybrid Cloud Console 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. 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 7.0 Update 2 or later
- Version 8.0 Update 1 or later

You can host the VMware vSphere infrastructure on-premise or on a VMware Cloud Verified provider that meets the requirements outlined in the following table:

Table 2.1. Version requirements for vSphere virtual environments

Virtual environment product	Required version
VMware virtual hardware	15 or later
vSphere ESXi hosts	7.0 Update 2 or later; 8.0 Update 1 or later
vCenter host	7.0 Update 2 or later; 8.0 Update 1 or later

IMPORTANT

Installing a cluster on VMware vSphere versions 7.0 and 7.0 Update 1 is deprecated. These versions are still fully supported, but all vSphere 6.x versions are no longer supported. Version 4.12 of OpenShift Container Platform requires VMware virtual hardware version 15 or later. To update the hardware version for your vSphere virtual machines, see the "Updating hardware on nodes running in vSphere" article in the *Updating clusters* section.

Table 2.2. Minimum supported vSphere version for VMware components

Component	Minimum supported versions	Description

Component	Minimum supported versions	Description
Hypervisor	vSphere 7.0 Update 2 (or later) or vSphere 8.0 Update 1 (or later) with virtual hardware version 15	This 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.
Storage with in-tree drivers	vSphere 7.0 Update 2 or later; 8.0 Update 1 or later	This plugin creates vSphere storage by using the in-tree storage drivers for vSphere included in OpenShift Container Platform.

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.

2.5. 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
	10256	openshift-sdn

Protocol	Port	Description
UDP	**4789**	virtual extensible LAN (VXLAN)
	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	IPsec Encapsulating Security Payload (ESP)

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

2.6. VMWARE VSPHERE CSI DRIVER OPERATOR REQUIREMENTS

To install the vSphere CSI Driver Operator, the following requirements must be met:

- VMware vSphere version: 7.0 Update 2 or later; 8.0 Update 1 or later
- vCenter version: 7.0 Update 2 or later; 8.0 Update 1 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.

Additional resources

- To remove a third-party 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.

2.7. VCENTER REQUIREMENTS

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

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. While most of the privileges are always required, some 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 amend vSphere roles for the specified objects to grant the required privileges.

An additional role is required if the installation program is 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 will be created in the cluster root	**Host.Config.Storage** **Resource.AssignVMToPool** **VApp.AssignResourcePool** **VApp.Import** **VirtualMachine.Config.AddNewDisk**
vSphere vCenter Resource Pool	If an existing resource pool is provided	**Host.Config.Storage** **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**

vSphere object for role	When required	Required privileges in vSphere API
		VirtualMachine.Config.Rename **VirtualMachine.Config.ResetGuestInfo** **VirtualMachine.Config.Resource** **VirtualMachine.Config.Settings** **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 Datacenter	If the installation program creates the virtual machine folder. For UPI, **VirtualMachine.Inventory.Create** and **VirtualMachine.Inventory.Delete** privileges are optional if your cluster does not use the Machine API.	**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.Rename VirtualMachine.Config.ResetGuestInfo VirtualMachine.Config.Resource VirtualMachine.Config.Settings 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.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" "Profile-driven storage"."Profile-driven storage update" "Profile-driven storage"."Profile-driven storage view"**
vSphere vCenter Cluster	If VMs will be created 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 an existing resource pool is provided	**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 object for role	When required	Required privileges in vCenter GUI
vSphere Datastore	Always	**Datastore."Allocate space" Datastore."Browse datastore" Datastore."Low level file operations" "vSphere Tagging"."Assign or Unassign vSphere Tag on Object"**
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**

vSphere object for role	When required	Required privileges in vCenter GUI
		Configuration".Rename "Virtual machine"."Change Configuration"."Reset guest information" "Virtual machine"."Change Configuration"."Change resource" "Virtual machine"."Change Configuration"."Change Settings" "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 Datacenter	If the installation program creates the virtual machine folder. For UPI, **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**

vSphere object for role	When required	Required privileges in vCenter GUI
		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 Settings" "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

vSphere object for role	When required	Required privileges in vCenter GUI
		Inventory"."Create from existing "Virtual machine"."Edit Inventory"."Remove" "Virtual machine".Provisioning."Clone virtual machine" "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 propogate 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 Datacenter	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

vSphere object	When required	Propagate to children	Permissions required

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

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.

- OpenShift Container Platform generally supports compute-only vMotion, 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.

- Using Storage vMotion can cause issues and is not supported. If you are using 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 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.

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 is destroyed during the cluster installation process. A minimum of 800 GB of storage is required to use a standard cluster.

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

Cluster limits

Available resources vary between clusters. The number of possible clusters within a vCenter is limited 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.

Networking requirements

You must use the Dynamic Host Configuration Protocol (DHCP) for the network and ensure that the DHCP server is configured to provide persistent IP addresses to the cluster machines. In the DHCP lease, you must configure the DHCP to use the default gateway. All nodes must be in the same VLAN. You cannot scale the cluster using a second VLAN as a Day 2 operation. Additionally, you must create the following networking resources before you install the OpenShift Container Platform cluster:

> **NOTE**
>
> It is recommended that each OpenShift Container Platform node in the cluster must have access to a Network Time Protocol (NTP) server that is discoverable via DHCP. Installation is possible without an NTP server. However, asynchronous server clocks will cause errors, which NTP server prevents.

Required IP Addresses

An installer-provisioned vSphere installation requires two static IP addresses:

- The **API** address is used to access the cluster API.
- The **Ingress** address is used for cluster ingress traffic.

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

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	**api.<cluster_name>.<base_domain>.**	This DNS A/AAAA or CNAME 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.

Component	Record	Description
Ingress VIP	***.apps.<cluster_name>.<base_domain>.**	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.

2.8. GENERATING A KEY PAIR FOR CLUSTER NODE SSH ACCESS

During an OpenShift Container Platform installation, you can provide an SSH public key to the installation program. The key is passed to the Red Hat Enterprise Linux CoreOS (RHCOS) nodes through their Ignition config files and is used to authenticate SSH access to the nodes. The key is added to the **~/.ssh/authorized_keys** list for the **core** user on each node, which enables password-less authentication.

After the key is passed to the nodes, 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 such as AWS key pairs.

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> 1
```

1 Specify 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 FIPS validated or Modules In Process cryptographic libraries on 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> (1)
```

(1) Specify 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.

2.9. 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 4.12 release notes* document.

Procedure

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

2. Select your infrastructure provider.

3. Navigate to the page for your installation type, download the installation program that corresponds with your host operating system and architecture, and place the file in the directory where you will store the installation configuration files.

   > **IMPORTANT**
   >
   > The installation program creates several files on the computer that you use to install your cluster. You must keep the installation program and the files that the installation program creates after you finish installing the cluster. Both files are required to delete the cluster.

   > **IMPORTANT**
   >
   > 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.

4. 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
   ```

5. Download your installation pull secret from the 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.

2.10. 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

1. 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.

2. 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
```

2.11. DEPLOYING THE CLUSTER

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

When you have configured your VMC environment for OpenShift Container Platform deployment, you use the OpenShift Container Platform installation program from the bastion management host that is co-located in the VMC environment. The installation program and control plane automates the process of deploying and managing the resources needed for the OpenShift Container Platform cluster.

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

Prerequisites

- Configure an account with the cloud platform that hosts your cluster.
- Obtain the OpenShift Container Platform installation program and the pull secret for your cluster.
- Verify 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.

Procedure

1. Change to the directory that contains the installation program and initialize the cluster deployment:

```
$ ./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 the Red Hat OpenShift Cluster Manager .

> **IMPORTANT**
>
> Use the **openshift-install** command from the bastion hosted in the VMC environment.

+

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

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.

2.12. INSTALLING THE OPENSHIFT CLI BY DOWNLOADING THE BINARY

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

IMPORTANT

If you installed an earlier version of **oc**, you cannot use it to complete all of the commands in OpenShift Container Platform 4.12. Download and install the new version of **oc**.

Installing the OpenShift CLI on Linux

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

Procedure

1. Navigate to the OpenShift Container Platform downloads page on the Red Hat Customer Portal.

2. Select the architecture from the **Product Variant** drop-down list.

3. Select the appropriate version from the **Version** drop-down list.

4. Click **Download Now** next to the **OpenShift v4.12 Linux Client** 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>
```

Installing the OpenShift CLI on Windows

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

Procedure

1. Navigate to the OpenShift Container Platform downloads page on the Red Hat Customer Portal.
2. Select the appropriate version from the **Version** drop-down list.
3. Click **Download Now** next to the **OpenShift v4.12 Windows Client** entry and save the file.
4. Unzip the archive with a ZIP program.
5. Move the **oc** binary to a directory that is on your **PATH**.
   To check your **PATH**, open the command prompt and execute the following command:

   ```
   C:\> path
   ```

Verification

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

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

Installing the OpenShift CLI on macOS

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

Procedure

1. Navigate to the OpenShift Container Platform downloads page on the Red Hat Customer Portal.
2. Select the appropriate version from the **Version** drop-down list.
3. Click **Download Now** next to the **OpenShift v4.12 macOS Client** entry and save the file.

   > **NOTE**
   >
   > For macOS arm64, choose the **OpenShift v4.12 macOS arm64 Client** entry.

4. Unpack and unzip the archive.
5. Move the **oc** binary to a directory on your PATH.
   To check your **PATH**, open a terminal and execute the following command:

   ```
   $ echo $PATH
   ```

Verification

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

  ```
  $ oc <command>
  ```

2.13. LOGGING IN TO THE CLUSTER BY USING THE CLI

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

Prerequisites

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

Procedure

1. Export the **kubeadmin** credentials:

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

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

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

   ```
   $ oc whoami
   ```

   Example output

   ```
   system:admin
   ```

2.14. CREATING REGISTRY STORAGE

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

2.14.1. 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.

2.14.2. 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.

Instructions are shown for configuring a persistent volume, which is required for production clusters. Where applicable, instructions are shown for configuring an empty directory as the storage location, which is available for only non-production clusters.

Additional instructions are provided for allowing the image registry to use block storage types by using the **Recreate** rollout strategy during upgrades.

2.14.2.1. 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. To configure your registry to use storage, change the **spec.storage.pvc** 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:

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

Example output

```
No resourses 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:

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

Example output

```
storage:
  pvc:
    claim: 1
```

1 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:

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

Example output

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

2.14.2.2. 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 PV for the block storage device, and create a 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 (1)
     namespace: openshift-image-registry (2)
   spec:
     accessModes:
     - ReadWriteOnce (3)
     resources:
       requests:
         storage: 100Gi (4)
   ```

   (1) A unique name that represents the **PersistentVolumeClaim** object.

   (2) The namespace for the **PersistentVolumeClaim** object, which is **openshift-image-registry**.

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

   (4) 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: (1)
   ```

   (1) 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.

2.15. BACKING UP VMWARE VSPHERE VOLUMES

OpenShift Container Platform provisions new volumes as independent persistent disks to freely attach and detach the volume on any node in the cluster. As a consequence, it is not possible to back up volumes that use snapshots, or to restore volumes from snapshots. See Snapshot Limitations for more information.

Procedure

To create a backup of persistent volumes:

1. Stop the application that is using the persistent volume.
2. Clone the persistent volume.
3. Restart the application.
4. Create a backup of the cloned volume.
5. Delete the cloned volume.

2.16. TELEMETRY ACCESS FOR OPENSHIFT CONTAINER PLATFORM

In OpenShift Container Platform 4.12, the Telemetry service, which runs by default to provide metrics about cluster health and the success of updates, requires internet access. If your cluster is connected to the internet, Telemetry runs automatically, and your cluster is registered to OpenShift Cluster Manager Hybrid Cloud Console.

After you confirm that your OpenShift Cluster Manager Hybrid Cloud Console 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.

Additional resources

- See About remote health monitoring for more information about the Telemetry service

2.17. SERVICES FOR AN EXTERNAL LOAD BALANCER

You can configure an OpenShift Container Platform cluster to use an external load balancer in place of the default load balancer.

IMPORTANT

Configuring an external 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 an external load balancer:

- Ingress Controller
- OpenShift API
- OpenShift MachineConfig API

You can choose whether you want to configure one or all of these services for an external 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

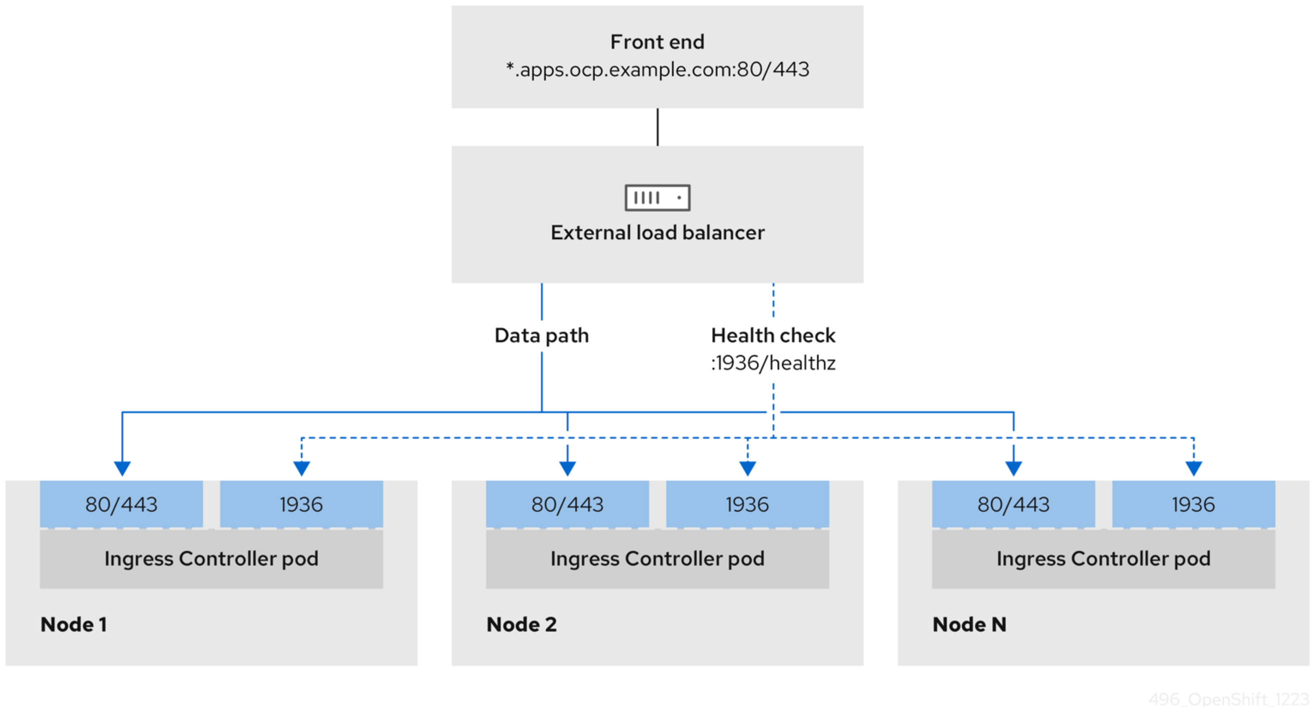


Figure 2.2. Example network workflow that shows an OpenShift API operating in an OpenShift Container Platform environment

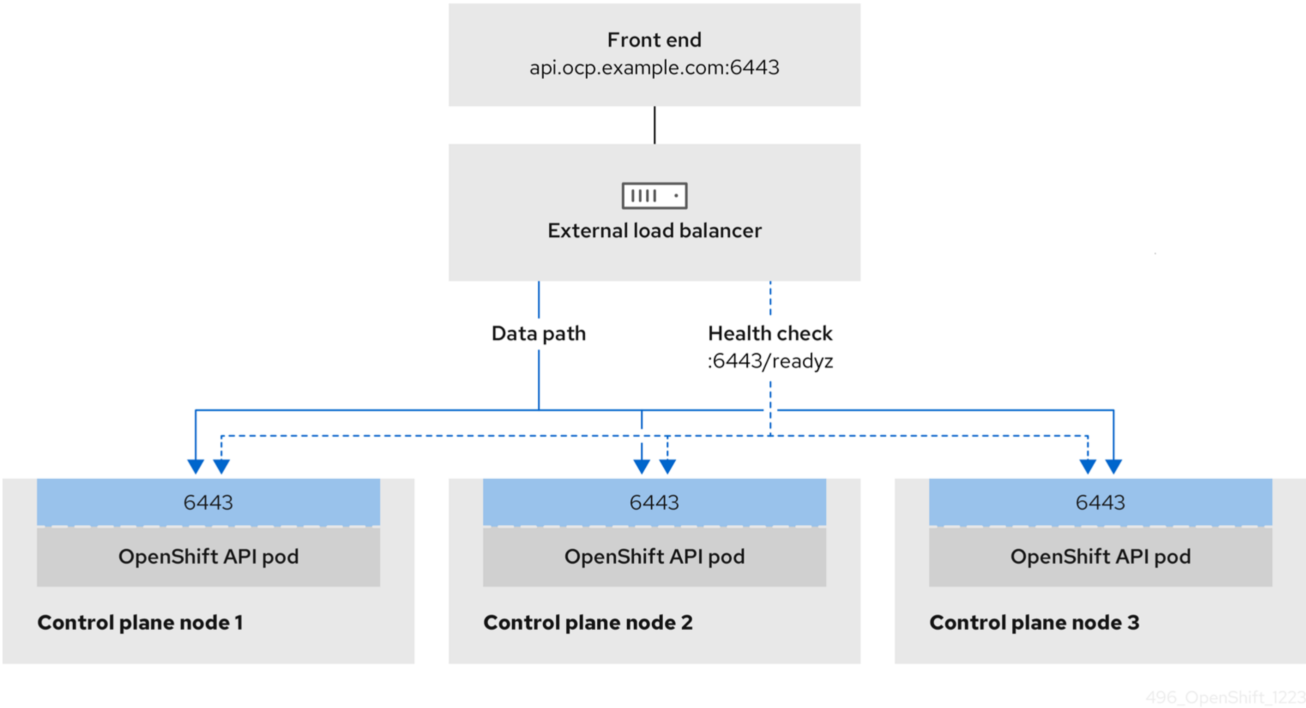
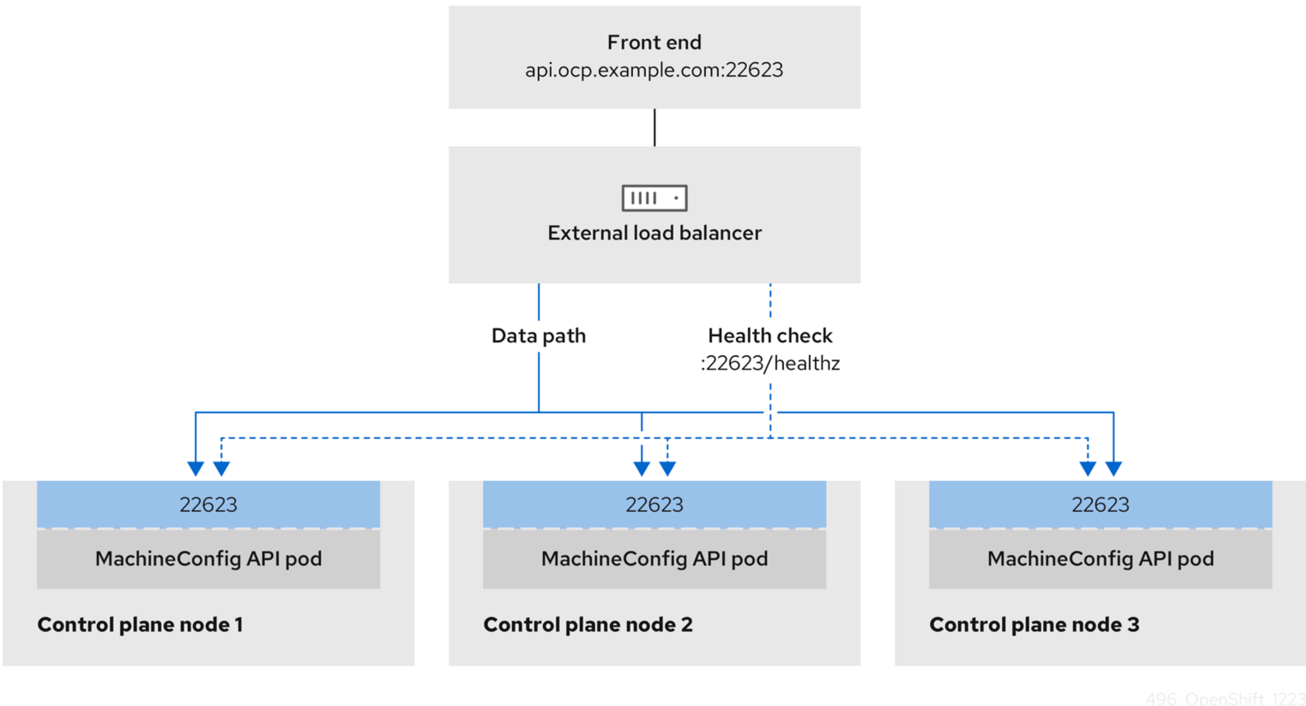


Figure 2.3. Example network workflow that shows an OpenShift MachineConfig API operating in an OpenShift Container Platform environment



The following configuration options are supported for external 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 an external 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 external 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 external 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.

2.17.1. Configuring an external load balancer

You can configure an OpenShift Container Platform cluster to use an external load balancer in place of the default load balancer.

IMPORTANT

Before you configure an external load balancer, ensure that you read the "Services for an external load balancer" section.

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

NOTE

MetalLB, that runs on a cluster, functions as an external load balancer.

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.

Ingress Controller prerequisites

- You defined a front-end IP address.
- TCP ports 443 and 80 are exposed on the front-end IP address of your load balancer.
- The front-end IP address, port 80 and port 443 are be 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 to 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.

Prerequisite 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 examples demonstrate health check specifications for the previously listed backend services:

Example of a Kubernetes API health check specification

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

Example of a Machine Config API health check specification

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

Example of an Ingress Controller health check specification

```
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, 443, and 80:

   Example HAProxy configuration

```
#...
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
# ...
```

2. Use the **curl** CLI command to verify that the external 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=UlYWOyQ62LWjw2h003xtYSKlh1a0Py2hhctw0WmV2YEdhJjFyQwWcGBsja261dG
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 external load balancer. You must update records to your DNS server for the cluster API and applications over the load balancer.

Examples of 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. Use the **curl** CLI command to verify that the external 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, 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/JuLJMeoKNXlfiVv2YgZc09c3TBOBU4NI6kDXaJH1LdicNhN1UsQ
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=UlYWOyQ62LWjw2h003xtYSKlh1a0Py2hhctw0WmV2YEdhJjFyQwWcGBsja261dG
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
```

2.18. NEXT STEPS

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

CHAPTER 3. INSTALLING A CLUSTER ON VMC WITH CUSTOMIZATIONS

In OpenShift Container Platform version 4.12, you can install a cluster on your VMware vSphere instance using installer-provisioned infrastructure by deploying it to VMware Cloud (VMC) on AWS.

Once you configure your VMC environment for OpenShift Container Platform deployment, you use the OpenShift Container Platform installation program from the bastion management host, co-located in the VMC environment. The installation program and control plane automates the process of deploying and managing the resources needed for the OpenShift Container Platform cluster.

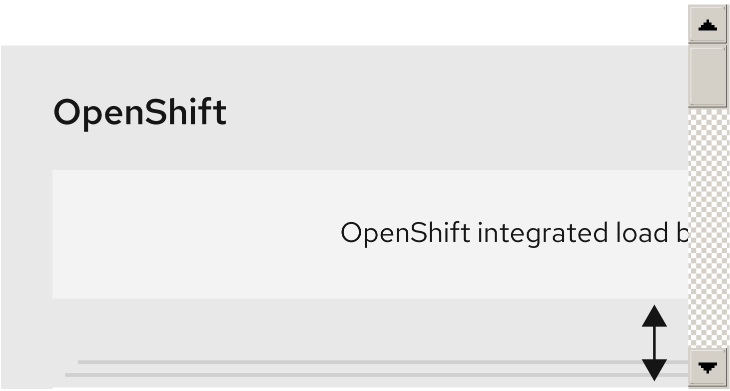
To customize the OpenShift Container Platform installation, you modify parameters in the **install-config.yaml** file before you install the cluster.

NOTE

OpenShift Container Platform supports deploying a cluster to a single VMware vCenter only. Deploying a cluster with machines/machine sets on multiple vCenters is not supported.

3.1. SETTING UP VMC FOR VSPHERE

You can install OpenShift Container Platform on VMware Cloud (VMC) on AWS hosted vSphere clusters to enable applications to be deployed and managed both on-premise and off-premise, across the hybrid cloud.



You must configure several options in your VMC environment prior to installing OpenShift Container Platform on VMware vSphere. Ensure your VMC environment has the following prerequisites:

- Create a non-exclusive, DHCP-enabled, NSX-T network segment and subnet. Other virtual machines (VMs) can be hosted on the subnet, but at least eight IP addresses must be available for the OpenShift Container Platform deployment.
- Allocate two IP addresses, outside the DHCP range, and configure them with reverse DNS records.
    - A DNS record for **api.<cluster_name>.<base_domain>** pointing to the allocated IP address.
    - A DNS record for ***.apps.<cluster_name>.<base_domain>** pointing to the allocated IP address.
- Configure the following firewall rules:

- An ANY:ANY firewall rule between the OpenShift Container Platform compute network and the internet. This is used by nodes and applications to download container images.
- An ANY:ANY firewall rule between the installation host and the software-defined data center (SDDC) management network on port 443. This allows you to upload the Red Hat Enterprise Linux CoreOS (RHCOS) OVA during deployment.
- An HTTPS firewall rule between the OpenShift Container Platform compute network and vCenter. This connection allows OpenShift Container Platform to communicate with vCenter for provisioning and managing nodes, persistent volume claims (PVCs), and other resources.

- You must have the following information to deploy OpenShift Container Platform:
  - The OpenShift Container Platform cluster name, such as **vmc-prod-1**.
  - The base DNS name, such as **companyname.com**.
  - If not using the default, the pod network CIDR and services network CIDR must be identified, which are set by default to **10.128.0.0/14** and **172.30.0.0/16**, respectively. These CIDRs are used for pod-to-pod and pod-to-service communication and are not accessible externally; however, they must not overlap with existing subnets in your organization.
  - The following vCenter information:
    - vCenter hostname, username, and password
    - Datacenter name, such as **SDDC-Datacenter**
    - Cluster name, such as **Cluster-1**
    - Network name
    - Datastore name, such as **WorkloadDatastore**

NOTE

It is recommended to move your vSphere cluster to the VMC **Compute-ResourcePool** resource pool after your cluster installation is finished.

- A Linux-based host deployed to VMC as a bastion.
  - The bastion host can be Red Hat Enterprise Linux (RHEL) or any another Linux-based host; it must have internet connectivity and the ability to upload an OVA to the ESXi hosts.
  - Download and install the OpenShift CLI tools to the bastion host.
    - The **openshift-install** installation program
    - The OpenShift CLI (**oc**) tool

NOTE

You cannot use the VMware NSX Container Plugin for Kubernetes (NCP), and NSX is not used as the OpenShift SDN. The version of NSX currently available with VMC is incompatible with the version of NCP certified with OpenShift Container Platform.

However, the NSX DHCP service is used for virtual machine IP management with the full-stack automated OpenShift Container Platform deployment and with nodes provisioned, either manually or automatically, by the Machine API integration with vSphere. Additionally, NSX firewall rules are created to enable access with the OpenShift Container Platform cluster and between the bastion host and the VMC vSphere hosts.

3.1.1. VMC Sizer tool

VMware Cloud on AWS is built on top of AWS bare metal infrastructure; this is the same bare metal infrastructure which runs AWS native services. When a VMware cloud on AWS software-defined data center (SDDC) is deployed, you consume these physical server nodes and run the VMware ESXi hypervisor in a single tenant fashion. This means the physical infrastructure is not accessible to anyone else using VMC. It is important to consider how many physical hosts you will need to host your virtual infrastructure.

To determine this, VMware provides the VMC on AWS Sizer. With this tool, you can define the resources you intend to host on VMC:

- Types of workloads
- Total number of virtual machines
- Specification information such as:
  - Storage requirements
  - vCPUs
  - vRAM
  - Overcommit ratios

With these details, the sizer tool can generate a report, based on VMware best practices, and recommend your cluster configuration and the number of hosts you will need.

3.2. VSPHERE PREREQUISITES

- 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 block registry storage. For more information on persistent storage, see Understanding persistent storage.
- 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. INTERNET ACCESS FOR OPENSHIFT CONTAINER PLATFORM

In OpenShift Container Platform 4.12, you require access to the internet to install your cluster.

You must have internet access to:

- Access OpenShift Cluster Manager Hybrid Cloud Console 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. 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 7.0 Update 2 or later
- Version 8.0 Update 1 or later

You can host the VMware vSphere infrastructure on-premise or on a VMware Cloud Verified provider that meets the requirements outlined in the following table:

Table 3.1. Version requirements for vSphere virtual environments

Virtual environment product	Required version
VMware virtual hardware	15 or later
vSphere ESXi hosts	7.0 Update 2 or later; 8.0 Update 1 or later
vCenter host	7.0 Update 2 or later; 8.0 Update 1 or later

IMPORTANT

Installing a cluster on VMware vSphere versions 7.0 and 7.0 Update 1 is deprecated. These versions are still fully supported, but all vSphere 6.x versions are no longer supported. Version 4.12 of OpenShift Container Platform requires VMware virtual hardware version 15 or later. To update the hardware version for your vSphere virtual machines, see the "Updating hardware on nodes running in vSphere" article in the *Updating clusters* section.

Table 3.2. Minimum supported vSphere version for VMware components

Component	Minimum supported versions	Description
Hypervisor	vSphere 7.0 Update 2 (or later) or vSphere 8.0 Update 1 (or later) with virtual hardware version 15	This 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.
Storage with in-tree drivers	vSphere 7.0 Update 2 or later; 8.0 Update 1 or later	This plugin creates vSphere storage by using the in-tree storage drivers for vSphere included in OpenShift Container Platform.

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.

3.5. 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 3.3. 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

Protocol	Port	Description
	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
	10256	openshift-sdn
UDP	**4789**	virtual extensible LAN (VXLAN)
	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	IPsec Encapsulating Security Payload (ESP)

Table 3.4. Ports used for all-machine to control plane communications

Protocol	Port	Description
TCP	**6443**	Kubernetes API

Table 3.5. Ports used for control plane machine to control plane machine communications

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

3.6. VMWARE VSPHERE CSI DRIVER OPERATOR REQUIREMENTS

To install the vSphere CSI Driver Operator, the following requirements must be met:

- VMware vSphere version: 7.0 Update 2 or later; 8.0 Update 1 or later
- vCenter version: 7.0 Update 2 or later; 8.0 Update 1 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.

Additional resources

- To remove a third-party 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.

3.7. VCENTER REQUIREMENTS

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

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. While most of the privileges are always required, some 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 amend vSphere roles for the specified objects to grant the required privileges.

An additional role is required if the installation program is to create a vSphere virtual machine folder.

Example 3.1. Roles and privileges required for installation in vSphere API

vSphere object for role	When required	Required privileges 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 will be created in the cluster root	**Host.Config.Storage Resource.AssignVMToPool VApp.AssignResourcePool VApp.Import VirtualMachine.Config.AddNewDisk**
vSphere vCenter Resource Pool	If an existing resource pool is provided	**Host.Config.Storage 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.Add**

vSphere object for role	When required	Required privileges in vSphere API
		RemoveDevice **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** **VirtualMachine.Config.ResetGuestInfo** **VirtualMachine.Config.Resource** **VirtualMachine.Config.Settings** **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 Datacenter	If the installation program creates the virtual machine folder. For UPI, **VirtualMachine.Inventory.Create** and **VirtualMachine.Inventory.Delete** privileges are optional if your cluster does not use the Machine API.	**InventoryService.Tagging.ObjectAttachable** **Resource.AssignVMToPool** **VApp.Import** **VirtualMachine.Config.AddExistingDisk** **VirtualMachine.Config.AddNewDisk**

vSphere object for role	When required	Required privileges in vSphere API
		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 VirtualMachine.Config.ResetGuestInfo VirtualMachine.Config.Resource VirtualMachine.Config.Settings 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.DeployTemplate VirtualMachine.Provisioning.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 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" "Profile-driven storage"."Profile-driven storage update" "Profile-driven storage"."Profile-driven storage view"**
vSphere vCenter Cluster	If VMs will be created 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 an existing resource pool is provided	**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 object for role	When required	Required privileges in vCenter GUI
vSphere Datastore	Always	**Datastore."Allocate space"** **Datastore."Browse datastore"** **Datastore."Low level file operations"** **"vSphere Tagging"."Assign or Unassign vSphere Tag on Object"**
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"**

vSphere object for role	When required	Required privileges in vCenter GUI
		"Virtual machine"."Change Configuration"."Rename" "Virtual machine"."Change Configuration"."Reset guest information" "Virtual machine"."Change Configuration"."Change resource" "Virtual machine"."Change Configuration"."Change Settings" "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 Datacenter	If the installation program creates the virtual machine folder. For UPI, **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"**

vSphere object for role	When required	Required privileges in vCenter GUI
		"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 Settings" "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"

vSphere object for role	When required	Required privileges in vCenter GUI
		"Virtual machine"."Edit Inventory"."Create from existing" **"Virtual machine"."Edit Inventory"."Remove"** **"Virtual machine".Provisioning."Clone virtual machine"** **"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 propogate 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 Datacenter	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 object	When required	Propagate to children	Permissions required
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.

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.

- OpenShift Container Platform generally supports compute-only vMotion, 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.

- Using Storage vMotion can cause issues and is not supported. If you are using 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 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.

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 is destroyed during the cluster installation process. A minimum of 800 GB of storage is required to use a standard cluster.

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

Cluster limits

Available resources vary between clusters. The number of possible clusters within a vCenter is limited 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.

Networking requirements

You must use the Dynamic Host Configuration Protocol (DHCP) for the network and ensure that the DHCP server is configured to provide persistent IP addresses to the cluster machines. In the DHCP lease, you must configure the DHCP to use the default gateway. All nodes must be in the same VLAN. You cannot scale the cluster using a second VLAN as a Day 2 operation. Additionally, you must create the following networking resources before you install the OpenShift Container Platform cluster:

> **NOTE**
>
> It is recommended that each OpenShift Container Platform node in the cluster must have access to a Network Time Protocol (NTP) server that is discoverable via DHCP. Installation is possible without an NTP server. However, asynchronous server clocks will cause errors, which NTP server prevents.

Required IP Addresses

An installer-provisioned vSphere installation requires two static IP addresses:

- The **API** address is used to access the cluster API.
- The **Ingress** address is used for cluster ingress traffic.

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

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 3.6. Required DNS records

Component	Record	Description
API VIP	**api.<cluster_name>.<base_domain>.**	This DNS A/AAAA or CNAME 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.

Component	Record	Description
Ingress VIP	***.apps.<cluster_name>.<base_domain>.**	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.

3.8. GENERATING A KEY PAIR FOR CLUSTER NODE SSH ACCESS

During an OpenShift Container Platform installation, you can provide an SSH public key to the installation program. The key is passed to the Red Hat Enterprise Linux CoreOS (RHCOS) nodes through their Ignition config files and is used to authenticate SSH access to the nodes. The key is added to the **~/.ssh/authorized_keys** list for the **core** user on each node, which enables password-less authentication.

After the key is passed to the nodes, 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 such as AWS key pairs.

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> 1
```

1 Specify 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 FIPS validated or Modules In Process cryptographic libraries on 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> 1
```

1 Specify 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.

3.9. 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 4.12 release notes* document.

Procedure

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

> **IMPORTANT**
>
> The installation program creates several files on the computer that you use to install your cluster. You must keep the installation program and the files that the installation program creates after you finish installing the cluster. Both files are required to delete the cluster.

> **IMPORTANT**
>
> 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.

4. 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
```

5. Download your installation pull secret from the 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.

3.10. 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

1. 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.

2. 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
```

3.11. VMWARE VSPHERE REGION AND ZONE ENABLEMENT

You can deploy an OpenShift Container Platform cluster to multiple vSphere datacenters that run in a single VMware vCenter. Each datacenter 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

VMware vSphere region and zone enablement 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 .

The default installation configuration deploys a cluster to a single vSphere datacenter. If you want to deploy a cluster to multiple vSphere datacenters, 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 datacenters 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 datacenter.

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

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

Table 3.7. Example of a configuration with multiple vSphere datacenters that run in a single VMware vCenter

Datacenter (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

3.12. CREATING THE INSTALLATION CONFIGURATION FILE

You can customize the OpenShift Container Platform cluster you install on VMware vSphere.

Prerequisites

- Obtain the OpenShift Container Platform installation program and the pull secret for your cluster.
- Obtain service principal permissions at the subscription level.

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> 1
   ```

   1 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.

vi. Select the default vCenter datastore to use.

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.

xiii. Paste the pull secret from the Red Hat OpenShift Cluster Manager .

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.

3.12.1. Installation configuration parameters

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

NOTE

After installation, you cannot modify these parameters in the **install-config.yaml** file.

3.12.1.1. Required configuration parameters

Required installation configuration parameters are described in the following table:

Table 3.8. Required parameters

Parameter	Description	Values
apiVersion	The API version for the **install-config.yaml** content. The current version is **v1**. The installation program may also support older API versions.	String
baseDomain	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.	A fully-qualified domain or subdomain name, such as **example.com**.
metadata	Kubernetes resource **ObjectMeta**, from which only the **name** parameter is consumed.	Object
metadata.name	The name of the cluster. DNS records for the cluster are all subdomains of **{{.metadata.name}}.{{.baseDomain}}**.	String of lowercase letters and hyphens (**-**), such as **dev**.

Parameter	Description	Values
platform	The configuration for the specific platform upon which to perform the installation: **alibabacloud**, **aws**, **baremetal**, **azure**, **gcp**, **ibmcloud**, **nutanix**, **openstack**, **ovirt**, **vsphere**, or **{}**. For additional information about **platform.<platform>** parameters, consult the table for your specific platform that follows.	Object
pullSecret	Get a pull secret from the Red Hat OpenShift Cluster Manager to authenticate downloading container images for OpenShift Container Platform components from services such as Quay.io.	`{ "auths":{ "cloud.openshift.com":{ "auth":"b3Blb=", "email":"you@example.com" }, "quay.io":{ "auth":"b3Blb=", "email":"you@example.com" } } }`

3.12.1.2. 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 provide different IP address blocks than the defaults.

Only IPv4 addresses are supported.

NOTE

Globalnet is not supported with Red Hat OpenShift Data Foundation disaster recovery solutions. For regional disaster recovery scenarios, ensure that you use a nonoverlapping range of private IP addresses for the cluster and service networks in each cluster.

Table 3.9. Network parameters

Parameter	Description	Values

Parameter	Description	Values
networking	The configuration for the cluster network.	Object NOTE You cannot modify parameters specified by the **networking** object after installation.
networking.networkType	The Red Hat OpenShift Networking network plugin to install.	Either **OpenShiftSDN** or **OVNKubernetes**. **OpenShiftSDN** is a CNI plugin for all-Linux networks. **OVNKubernetes** is a CNI plugin for Linux networks and hybrid networks that contain both Linux and Windows servers. The default value is **OVNKubernetes**.

networking.clusterNetwork	The IP address blocks for pods.
	
	The default value is **10.128.0.0/14** with a host prefix of **/23**.
	
	If you specify multiple IP address blocks, the blocks must not overlap.
		An array of objects. For example:
		<pre>networking:
		  clusterNetwork:
		  - cidr: 10.128.0.0/14
		    hostPrefix: 23</pre>

networking.clusterNetwork.cidr	Required if you use **networking.clusterNetwork**. An IP address block. An IPv4 network.	An IP address block in Classless Inter-Domain Routing (CIDR) notation. The prefix length for an IPv4 block is between **0** and **32**.
networking.clusterNetwork.hostPrefix	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.	A subnet prefix. The default value is **23**.

networking.serviceNetwork	The IP address block for services. The default value is **172.30.0.0/16**.
	
	The OpenShift SDN and OVN-Kubernetes network plugins support only a single IP address block for the service network.
		An array with an IP address block in CIDR format. For example:
		<pre>networking:
		  serviceNetwork:
		   - 172.30.0.0/16</pre>

Parameter	Description	Values

networking.machineNetwork	The IP address blocks for machines.
	
	If you specify multiple IP address blocks, the blocks must not overlap.
		An array of objects. For example:
		
		`networking:`
		`  machineNetwork:`
		`  - cidr: 10.0.0.0/16`

networking.machineNetwork.cidr	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. For libvirt, the default value is **192.168.126.0/24**.	An IP network block in CIDR notation. For example, **10.0.0.0/16**. NOTE Set the **networking.machineNetwork** to match the CIDR that the preferred NIC resides in.

3.12.1.3. Optional configuration parameters

Optional installation configuration parameters are described in the following table:

Table 3.10. Optional parameters

Parameter	Description	Values
additionalTrustBundle	A PEM-encoded X.509 certificate bundle that is added to the nodes' trusted certificate store. This trust bundle may also be used when a proxy has been configured.	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 *Installing*.	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**.	String

Parameter	Description	Values
capabilities.additionalEnabledCapabilities	Extends the set of optional capabilities beyond what you specify in **baselineCapabilitySet**. You may specify multiple capabilities in this parameter.	String array
compute	The configuration for the machines that comprise the compute nodes.	Array of **MachinePool** objects.
compute.architecture	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).	String
compute.hyperthreading	Whether to enable or disable simultaneous multithreading, or **hyperthreading**, on compute machines. By default, simultaneous multithreading is enabled to increase the performance of your machines' cores. IMPORTANT If you disable simultaneous multithreading, ensure that your capacity planning accounts for the dramatically decreased machine performance.	**Enabled** or **Disabled**
compute.name	Required if you use **compute**. The name of the machine pool.	**worker**
compute.platform	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.	**alibabacloud**, **aws**, **azure**, **gcp**, **ibmcloud**, **nutanix**, **openstack**, **ovirt**, **vsphere**, or **{}**
compute.replicas	The number of compute machines, which are also known as worker machines, to provision.	A positive integer greater than or equal to **2**. The default value is **3**.

Parameter	Description	Values
featureSet	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".	String. The name of the feature set to enable, such as **TechPreviewNoUpgrade**.
controlPlane	The configuration for the machines that comprise the control plane.	Array of **MachinePool** objects.
controlPlane.architecture	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).	String
controlPlane.hyperthreading	Whether to enable or disable simultaneous multithreading, or **hyperthreading**, on control plane machines. By default, simultaneous multithreading is enabled to increase the performance of your machines' cores. IMPORTANT If you disable simultaneous multithreading, ensure that your capacity planning accounts for the dramatically decreased machine performance.	**Enabled** or **Disabled**
controlPlane.name	Required if you use **controlPlane**. The name of the machine pool.	**master**
controlPlane.platform	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.	**alibabacloud**, **aws**, **azure**, **gcp**, **ibmcloud**, **nutanix**, **openstack**, **ovirt**, **vsphere**, or **{}**
controlPlane.replicas	The number of control plane machines to provision.	The only supported value is **3**, which is the default value.

Parameter	Description	Values
credentialsMode	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. NOTE Not all CCO modes are supported for all cloud providers. For more information about CCO modes, see the *Cloud Credential Operator* entry in the *Cluster Operators reference* content. NOTE If your AWS account has service control policies (SCP) enabled, you must configure the **credentialsMode** parameter to **Mint**, **Passthrough** or **Manual**.	**Mint**, **Passthrough**, **Manual** or an empty string (**""**).

Parameter	Description	Values
fips	Enable or disable FIPS mode. The default is **false** (disabled). 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 Installing the system in FIPS mode. The use of FIPS validated or Modules In Process cryptographic libraries is only supported on OpenShift Container Platform deployments on the **x86_64**, **ppc64le**, and **s390x** architectures. **NOTE** If you are using Azure File storage, you cannot enable FIPS mode.	**false** or **true**
imageContentSources	Sources and repositories for the release-image content.	Array of objects. Includes a **source** and, optionally, **mirrors**, as described in the following rows of this table.
imageContentSources.source	Required if you use **imageContentSources**. Specify the repository that users refer to, for example, in image pull specifications.	String

Parameter	Description	Values
imageContentSources.mirrors	Specify one or more repositories that may also contain the same images.	Array of strings
publish	How to publish or expose the user-facing endpoints of your cluster, such as the Kubernetes API, OpenShift routes.	**Internal** or **External**. The default value is **External**. Setting this field to **Internal** is not supported on non-cloud platforms. IMPORTANT If the value of the field is set to **Internal**, the cluster will become non-functional. For more information, refer to BZ#1953035.
sshKey	The SSH key to authenticate access to 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.	For example, **sshKey: ssh-ed25519 AAAA..**.

3.12.1.4. Additional VMware vSphere configuration parameters

Additional VMware vSphere configuration parameters are described in the following table.

NOTE

The **platform.vsphere** parameter prefixes each parameter listed in the table.

Table 3.11. Additional VMware vSphere cluster parameters

Parameter	Description	Values
vCenter	The fully-qualified hostname or IP address of the vCenter server.	String

Parameter	Description	Values
username	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.	String
password	The password for the vCenter user name.	String
datacenter	The name of the data center to use in the vCenter instance.	String
defaultDatastore	The name of the default datastore to use for provisioning volumes.	String
folder	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.	String, for example, **/<datacenter_name>/vm/<folder_name>/<subfolder_name>**.
resourcePool	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 **/<datacenter_name>/host/<cluster_name>/Resources**.	String, for example, **/<datacenter_name>/host/<cluster_name>/Resources/<resource_pool_name>/<optional_nested_resource_pool_name>**.
network	The network in the vCenter instance that contains the virtual IP addresses and DNS records that you configured.	String
cluster	The vCenter cluster to install the OpenShift Container Platform cluster in.	String
apiVIPs	The virtual IP (VIP) address that you configured for control plane API access. NOTE 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.	An IP address, for example **128.0.0.1**.

Parameter	Description	Values
ingressVIPs	The virtual IP (VIP) address that you configured for cluster ingress. NOTE 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.	An IP address, for example **128.0.0.1**.
diskType	Optional. The disk provisioning method. This value defaults to the vSphere default storage policy if not set.	Valid values are **thin**, **thick**, or **eagerZeroedThick**.

3.12.1.5. Optional VMware vSphere machine pool configuration parameters

Optional VMware vSphere machine pool configuration parameters are described in the following table.

> NOTE
>
> The **platform.vsphere** parameter prefixes each parameter listed in the table.

Table 3.12. Optional VMware vSphere machine pool parameters

Parameter	Description	Values
clusterOSImage	The location from which the installation program downloads the RHCOS image. You must set this parameter to perform an installation in a restricted network.	An HTTP or HTTPS URL, optionally with a SHA-256 checksum. For example, **https://mirror.openshift.com/images/rhcos-<version>-vmware.<architecture>.ova**.
osDisk.diskSizeGB	The size of the disk in gigabytes.	Integer
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.	Integer

Parameter	Description	Values
coresPerSocket	The number of cores per socket in a virtual machine. The number of virtual sockets on the virtual machine is **platform.vsphere.cpus**/**platform.vsphere.coresPerSocket**. The default value for control plane nodes and worker nodes is **4** and **2**, respectively.	Integer
memoryMB	The size of a virtual machine's memory in megabytes.	Integer

3.12.1.6. Region and zone enablement configuration parameters

To use the region and zone enablement feature, you must specify region and zone enablement parameters in your installation file.

IMPORTANT

Before you modify the **install-config.yaml** file to configure a region and zone enablement environment, read the "VMware vSphere region and zone enablement" and the "Configuring regions and zones for a VMware vCenter" sections.

NOTE

The **platform.vsphere** parameter prefixes each parameter listed in the table.

Table 3.13. Region and zone enablement parameters

Parameter	Description	Values
failureDomains	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.	String
failureDomains.name	The name of the failure domain. The machine pools use this name to reference the failure domain.	String
failureDomains.server	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.	String
failureDomains.region	You define a region by using a tag from the **openshift-region** tag category. The tag must be attached to the vCenter datacenter.	String

Parameter	Description	Values
failureDomains.zone	You define a zone by using a tag from the **openshift-zone** tag category. The tag must be attached to the vCenter datacenter.	String
failureDomains.topology.computeCluster	This parameter defines the compute cluster associated with the failure domain. If you do not define this parameter in your configuration, the compute cluster takes the value of **platform.vsphere.cluster** and **platform.vsphere.datacenter**.	String
failureDomains.topology.folder	The absolute path of an existing folder where the installation program creates the virtual machines. If you do not define this parameter in your configuration, the folder takes the value of **platform.vsphere.folder**.	String
failureDomains.topology.datacenter	Defines the datacenter where OpenShift Container Platform virtual machines (VMs) operate. If you do not define this parameter in your configuration, the datacenter defaults to **platform.vsphere.datacenter**.	String
failureDomains.topology.datastore	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.	String
failureDomains.topology.networks	Lists any network in the vCenter instance that contains the virtual IP addresses and DNS records that you configured. If you do not define this parameter in your configuration, the network takes the value of **platform.vsphere.network**.	String
failureDomains.topology.resourcePool	Optional: The absolute path of an existing resource pool where the installation program creates the virtual machines, for example, **/<datacenter_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 **/<datacenter_name>/host/<cluster_name>/Resources**.	String

3.12.2. 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 modify the values of the required parameters.

```
apiVersion: v1
baseDomain: example.com 1
compute: 2
  name: worker
  replicas: 3
  platform:
    vsphere: 3
      cpus: 2
      coresPerSocket: 2
      memoryMB: 8192
      osDisk:
        diskSizeGB: 120
controlPlane: 4
  name: master
  replicas: 3
  platform:
    vsphere: 5
      cpus: 4
      coresPerSocket: 2
      memoryMB: 16384
      osDisk:
        diskSizeGB: 120
metadata:
  name: cluster 6
platform:
  vsphere:
    vcenter: your.vcenter.server
    username: username
    password: password
    datacenter: datacenter
    defaultDatastore: datastore
    folder: folder
    resourcePool: resource_pool 7
    diskType: thin 8
    network: VM_Network
    cluster: vsphere_cluster_name 9
    apiVIPs:
      - api_vip
    ingressVIPs:
      - ingress_vip
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 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. Only one control plane pool is used.

3 5 Optional: Provide additional configuration for the machine pool parameters for the compute and control plane machines.

6 The cluster name that you specified in your DNS records.

7 Optional: Provide 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.

8 The vSphere disk provisioning method.

9 The vSphere cluster to install the OpenShift Container Platform cluster in.

> **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.

3.12.3. Configuring the cluster-wide proxy during installation

Production environments can deny direct access to the internet and instead have an HTTP or HTTPS proxy available. You can configure a new OpenShift Container Platform cluster to use a proxy by configuring the proxy settings in the **install-config.yaml** file.

Prerequisites

- You have an existing **install-config.yaml** file.
- You 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 Platform (GCP), 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> 1
  httpsProxy: https://<username>:<pswd>@<ip>:<port> 2
  noProxy: example.com 3
```

```
additionalTrustBundle: | 4
    -----BEGIN CERTIFICATE-----
    <MY_TRUSTED_CA_CERT>
    -----END CERTIFICATE-----
additionalTrustBundlePolicy: <policy_to_add_additionalTrustBundle> 5
```

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

2. A proxy URL to use for creating HTTPS connections outside the cluster.

3. 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.

4. If provided, the installation program generates a config map that is named **user-ca-bundle** in the **openshift-config** namespace that contains one or more additional CA certificates that are required for proxying HTTPS connections. The Cluster Network Operator then creates a **trusted-ca-bundle** config map that merges these contents with the Red Hat Enterprise Linux CoreOS (RHCOS) trust bundle, and this config map is referenced in the **trustedCA** field of the **Proxy** object. The **additionalTrustBundle** field is required unless the proxy's identity certificate is signed by an authority from the RHCOS trust bundle.

5. Optional: The policy to determine 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**.

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.

3.12.4. Configuring regions and zones for a VMware vCenter

You can modify the default installation configuration file to deploy an OpenShift Container Platform cluster to multiple vSphere datacenters that run in a single VMware vCenter.

IMPORTANT

VMware vSphere region and zone enablement 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 .

IMPORTANT

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

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 datacenter objects for your VMware vCenter server. Consider specifying multiple failure domains if you need to provision virtual machine nodes in different datacenters, clusters, datastores, and other components. To enable regions and zones, you must define multiple failure domains for your OpenShift Container Platform cluster.

  NOTE

  You cannot change a failure domain after you installed an OpenShift Container Platform cluster on the VMware vSphere platform. You can add additional failure domains after cluster installation.

Procedure

1. Enter the following **govc** command-line tool commands to create the **openshift-region** and **openshift-zone** vCenter tag categories:

   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. To create a region tag for each region vSphere datacenter where you want to deploy your cluster, enter the following command in your terminal:

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

3. To create a zone tag for each vSphere cluster where you want to deploy your cluster, enter the following command:

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

4. Attach region tags to each vCenter datacenter object by entering the following command:

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

5. Attach the zone tags to each vCenter datacenter object by entering the following command:

```
$ govc tags.attach -c <zone_tag_category> <zone_tag_1> /<datacenter_1>/host/vcs-mdcnc-workload-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 datacenters defined in a vSphere center

```
apiVersion: v1
baseDomain: example.com
featureSet: TechPreviewNoUpgrade 1
compute:
  name: worker
  replicas: 3
  vsphere:
    zones: 2
      - "<machine_pool_zone_1>"
      - "<machine_pool_zone_2>"
controlPlane:
  name: master
  replicas: 3
  vsphere:
    zones: 3
      - "<machine_pool_zone_1>"
      - "<machine_pool_zone_2>"
metadata:
  name: cluster
platform:
  vsphere:
    vcenter: <vcenter_server> 4
    username: <username> 5
    password: <password> 6
```

```
    datacenter: datacenter 7
    defaultDatastore: datastore 8
    folder: "/<datacenter_name>/vm/<folder_name>/<subfolder_name>" 9
    cluster: cluster 10
    resourcePool: "/<datacenter_name>/host/<cluster_name>/Resources/<resource_pool_name>" 11
    diskType: thin
    failureDomains: 12
    - name: <machine_pool_zone_1> 13
      region: <region_tag_1> 14
      zone: <zone_tag_1> 15
      topology: 16
        datacenter: <datacenter1> 17
        computeCluster: "/<datacenter1>/host/<cluster1>" 18
        resourcePool: "/<datacenter1>/host/<cluster1>/Resources/<resourcePool1>" 19
        networks: 20
        - <VM_Network1_name>
        datastore: "/<datacenter1>/datastore/<datastore1>" 21
    - name: <machine_pool_zone_2>
      region: <region_tag_2>
      zone: <zone_tag_2>
      topology:
        datacenter: <datacenter2>
        computeCluster: "/<datacenter2>/host/<cluster2>"
        networks:
        - <VM_Network2_name>
        datastore: "/<datacenter2>/datastore/<datastore2>"
        resourcePool: "/<datacenter2>/host/<cluster2>/Resources/<resourcePool2>"
        folder: "/<datacenter2>/vm/<folder2>"
# ...
```

1 You must define set the **TechPreviewNoUpgrade** as the value for this parameter, so that you can use the VMware vSphere region and zone enablement feature.

2 3 An optional parameter for specifying a vCenter cluster. You define a zone by using a tag from the **openshift-zone** tag category. If you do not define this parameter, nodes will be distributed among all defined failure-domains.

4 5 6 7 8 9 10 11 The default vCenter topology. The installation program uses this topology information to deploy the bootstrap node. Additionally, the topology defines the default datastore for vSphere persistent volumes.

12 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. If you do not define this parameter, the installation program uses the default vCenter topology.

13 Defines the name of the failure domain. Each failure domain is referenced in the **zones** parameter to scope a machine pool to the failure domain.

14 You define a region by using a tag from the **openshift-region** tag category. The tag must be attached to the vCenter datacenter.

15 You define a zone by using a tag from the **openshift-zone tag** category. The tag must be attached to the vCenter datacenter.

16 Specifies the vCenter resources associated with the failure domain.

17 An optional parameter for defining the vSphere datacenter that is associated with a failure domain. If you do not define this parameter, the installation program uses the default vCenter topology.

18 An optional parameter for stating the absolute file path for the compute cluster that is associated with the failure domain. If you do not define this parameter, the installation program uses the default vCenter topology.

19 An optional parameter for the installer-provisioned infrastructure. The parameter sets the absolute path of an existing resource pool where the installation program creates the virtual machines, for example, **/<datacenter_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_datacenter/host/example_cluster/Resources**.

20 An optional parameter that lists any network in the vCenter instance that contains the virtual IP addresses and DNS records that you configured. If you do not define this parameter, the installation program uses the default vCenter topology.

21 An optional parameter for specifying a datastore to use for provisioning volumes. If you do not define this parameter, the installation program uses the default vCenter topology.

3.13. DEPLOYING THE CLUSTER

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

When you have configured your VMC environment for OpenShift Container Platform deployment, you use the OpenShift Container Platform installation program from the bastion management host that is co-located in the VMC environment. The installation program and control plane automates the process of deploying and managing the resources needed for the OpenShift Container Platform cluster.

IMPORTANT

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

Prerequisites

- Configure an account with the cloud platform that hosts your cluster.
- Obtain the OpenShift Container Platform installation program and the pull secret for your cluster.
- Verify 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.

Procedure

- Change to the directory that contains the installation program and initialize the cluster deployment:

```
$ ./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**.

> **IMPORTANT**
>
> Use the **openshift-install** command from the bastion hosted in the VMC environment.

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

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.

3.14. INSTALLING THE OPENSHIFT CLI BY DOWNLOADING THE BINARY

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

IMPORTANT

If you installed an earlier version of **oc**, you cannot use it to complete all of the commands in OpenShift Container Platform 4.12. Download and install the new version of **oc**.

Installing the OpenShift CLI on Linux

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

Procedure

1. Navigate to the OpenShift Container Platform downloads page on the Red Hat Customer Portal.

2. Select the architecture from the **Product Variant** drop-down list.

3. Select the appropriate version from the **Version** drop-down list.

4. Click **Download Now** next to the **OpenShift v4.12 Linux Client** 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>
```

Installing the OpenShift CLI on Windows

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

Procedure

1. Navigate to the OpenShift Container Platform downloads page on the Red Hat Customer Portal.
2. Select the appropriate version from the **Version** drop-down list.
3. Click **Download Now** next to the **OpenShift v4.12 Windows Client** entry and save the file.
4. Unzip the archive with a ZIP program.
5. Move the **oc** binary to a directory that is on your **PATH**.
   To check your **PATH**, open the command prompt and execute the following command:

```
C:\> path
```

Verification

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

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

Installing the OpenShift CLI on macOS

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

Procedure

1. Navigate to the OpenShift Container Platform downloads page on the Red Hat Customer Portal.
2. Select the appropriate version from the **Version** drop-down list.
3. Click **Download Now** next to the **OpenShift v4.12 macOS Client** entry and save the file.

> **NOTE**
>
> For macOS arm64, choose the **OpenShift v4.12 macOS arm64 Client** entry.

4. Unpack and unzip the archive.
5. Move the **oc** binary to a directory on your PATH.
   To check your **PATH**, open a terminal and execute the following command:

```
$ echo $PATH
```

Verification

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

```
$ oc <command>
```

3.15. LOGGING IN TO THE CLUSTER BY USING THE CLI

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

Prerequisites

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

Procedure

1. Export the **kubeadmin** credentials:

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

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

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

```
$ oc whoami
```

Example output

```
system:admin
```

3.16. CREATING REGISTRY STORAGE

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

3.16.1. 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.

3.16.2. 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.

Instructions are shown for configuring a persistent volume, which is required for production clusters. Where applicable, instructions are shown for configuring an empty directory as the storage location, which is available for only non-production clusters.

Additional instructions are provided for allowing the image registry to use block storage types by using the **Recreate** rollout strategy during upgrades.

3.16.2.1. 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. To configure your registry to use storage, change the **spec.storage.pvc** 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:

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

Example output

```
No resourses 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:

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

Example output

```
storage:
  pvc:
    claim: 1
```

1 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:

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

Example output

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

3.16.2.2. 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 PV for the block storage device, and create a 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 1
  namespace: openshift-image-registry 2
spec:
  accessModes:
  - ReadWriteOnce 3
  resources:
    requests:
      storage: 100Gi 4
```

1. A unique name that represents the **PersistentVolumeClaim** object.
2. The namespace for the **PersistentVolumeClaim** object, which is **openshift-image-registry**.
3. The access mode of the persistent volume claim. With **ReadWriteOnce**, the volume can be mounted with read and write permissions by a single node.
4. 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: 1
```

1. 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.

3.17. BACKING UP VMWARE VSPHERE VOLUMES

OpenShift Container Platform provisions new volumes as independent persistent disks to freely attach and detach the volume on any node in the cluster. As a consequence, it is not possible to back up volumes that use snapshots, or to restore volumes from snapshots. See Snapshot Limitations for more information.

Procedure

To create a backup of persistent volumes:

1. Stop the application that is using the persistent volume.
2. Clone the persistent volume.
3. Restart the application.
4. Create a backup of the cloned volume.
5. Delete the cloned volume.

3.18. TELEMETRY ACCESS FOR OPENSHIFT CONTAINER PLATFORM

In OpenShift Container Platform 4.12, the Telemetry service, which runs by default to provide metrics about cluster health and the success of updates, requires internet access. If your cluster is connected to the internet, Telemetry runs automatically, and your cluster is registered to OpenShift Cluster Manager Hybrid Cloud Console.

After you confirm that your OpenShift Cluster Manager Hybrid Cloud Console 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.

Additional resources

- See About remote health monitoring for more information about the Telemetry service

3.19. SERVICES FOR AN EXTERNAL LOAD BALANCER

You can configure an OpenShift Container Platform cluster to use an external load balancer in place of the default load balancer.

IMPORTANT

Configuring an external 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 an external load balancer:

- Ingress Controller
- OpenShift API
- OpenShift MachineConfig API

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

Figure 3.1. Example network workflow that shows an Ingress Controller operating in an OpenShift Container Platform environment

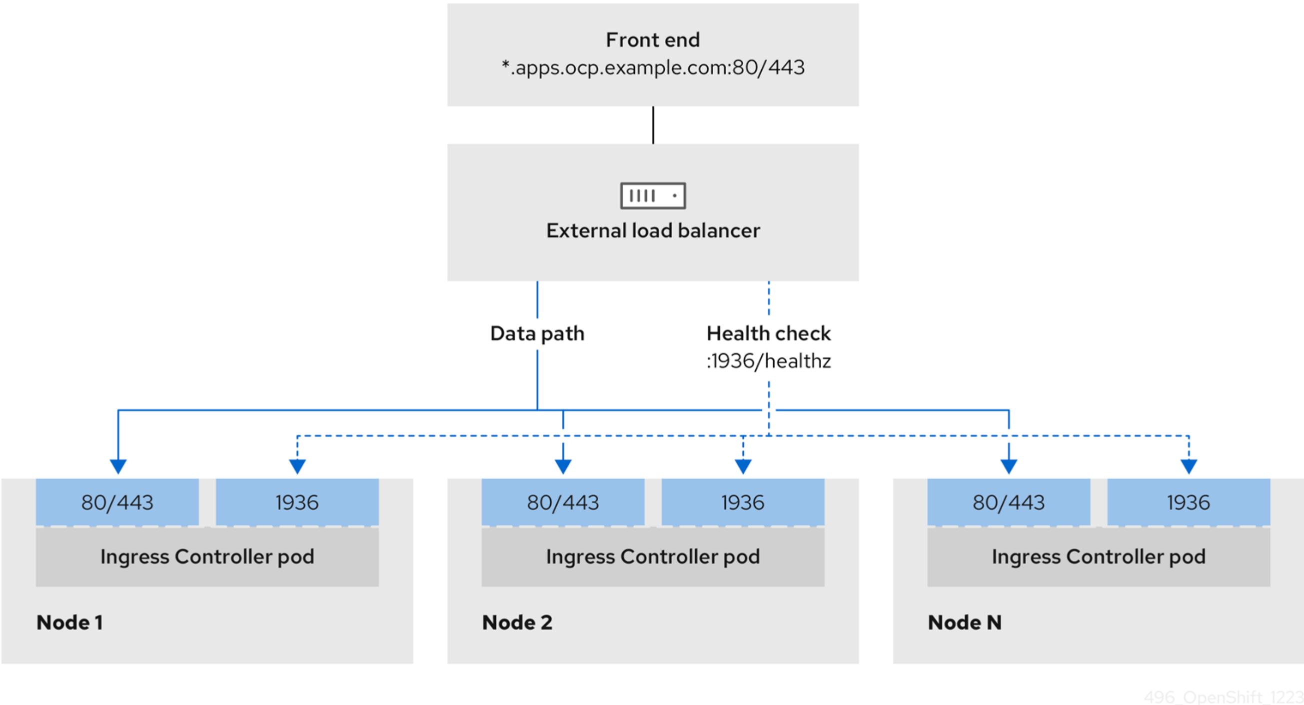


Figure 3.2. Example network workflow that shows an OpenShift API operating in an OpenShift Container Platform environment

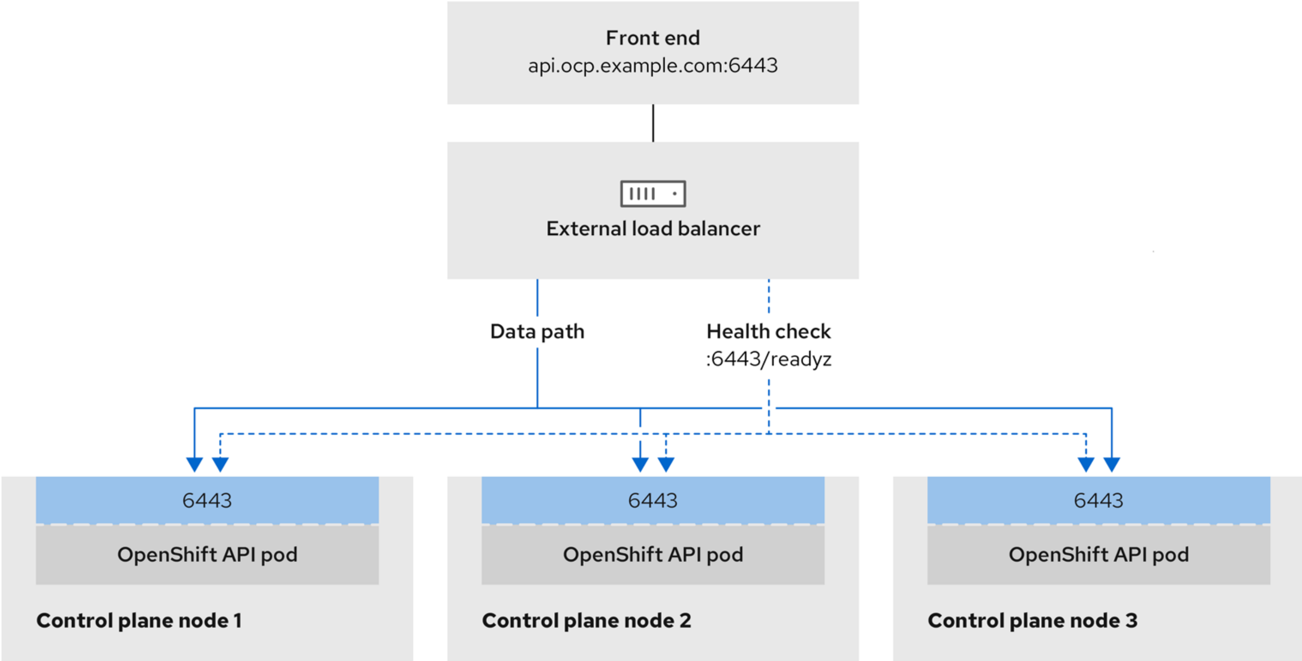
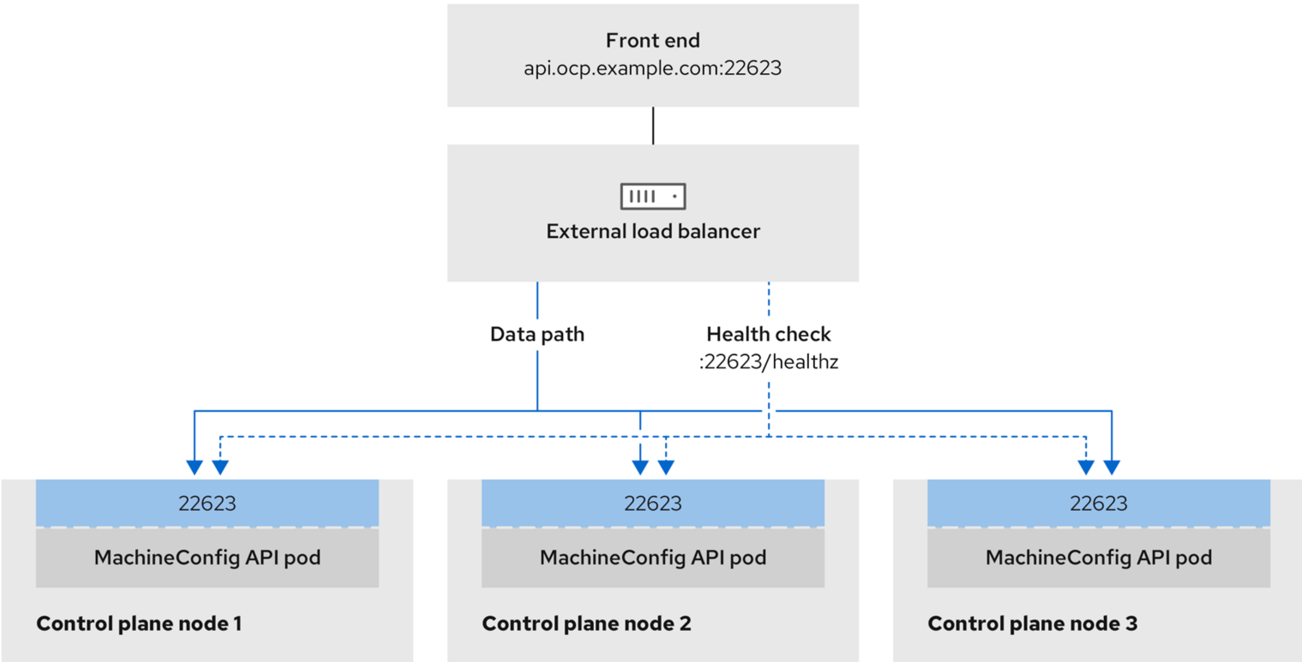


Figure 3.3. Example network workflow that shows an OpenShift MachineConfig API operating in an OpenShift Container Platform environment



The following configuration options are supported for external 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 an external 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 external 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 external 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.

3.19.1. Configuring an external load balancer

You can configure an OpenShift Container Platform cluster to use an external load balancer in place of the default load balancer.

IMPORTANT

Before you configure an external load balancer, ensure that you read the "Services for an external load balancer" section.

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

NOTE

MetalLB, that runs on a cluster, functions as an external load balancer.

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.

Ingress Controller prerequisites

- You defined a front-end IP address.
- TCP ports 443 and 80 are exposed on the front-end IP address of your load balancer.
- The front-end IP address, port 80 and port 443 are be 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 to 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.

Prerequisite 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 examples demonstrate health check specifications for the previously listed backend services:

Example of a Kubernetes API health check specification

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

Example of a Machine Config API health check specification

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

Example of an Ingress Controller health check specification

```
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, 443, and 80:

Example HAProxy configuration

```
#...
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
# ...
```

2. Use the **curl** CLI command to verify that the external 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=UlYWOyQ62LWjw2h003xtYSKlh1a0Py2hhctw0WmV2YEdhJjFyQwWcGBsja261dG
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 external load balancer. You must update records to your DNS server for the cluster API and applications over the load balancer.

Examples of 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. Use the **curl** CLI command to verify that the external 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, 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/JuLJMeoKNXlfiVv2YgZc09c3TBOBU4NI6kDXaJH1LdicNhN1UsQ
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=UlYWOyQ62LWjw2h003xtYSKlh1a0Py2hhctw0WmV2YEdhJjFyQwWcGBsja261dG
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
```

3.20. NEXT STEPS

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

CHAPTER 4. INSTALLING A CLUSTER ON VMC WITH NETWORK CUSTOMIZATIONS

In OpenShift Container Platform version 4.12, you can install a cluster on your VMware vSphere instance using installer-provisioned infrastructure with customized network configuration options by deploying it to VMware Cloud (VMC) on AWS.

Once you configure your VMC environment for OpenShift Container Platform deployment, you use the OpenShift Container Platform installation program from the bastion management host, co-located in the VMC environment. The installation program and control plane automates the process of deploying and managing the resources needed for the OpenShift Container Platform cluster.

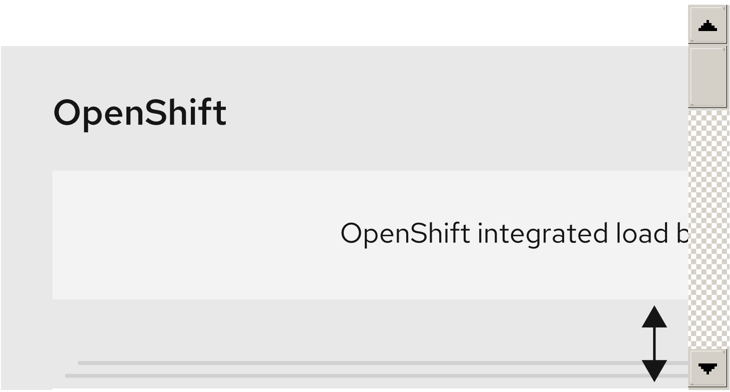
By customizing your OpenShift Container Platform network configuration, your cluster can coexist with existing IP address allocations in your environment and integrate with existing VXLAN configurations. To customize the installation, you modify parameters in the **install-config.yaml** file before you 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.

NOTE

OpenShift Container Platform supports deploying a cluster to a single VMware vCenter only. Deploying a cluster with machines/machine sets on multiple vCenters is not supported.

4.1. SETTING UP VMC FOR VSPHERE

You can install OpenShift Container Platform on VMware Cloud (VMC) on AWS hosted vSphere clusters to enable applications to be deployed and managed both on-premise and off-premise, across the hybrid cloud.



You must configure several options in your VMC environment prior to installing OpenShift Container Platform on VMware vSphere. Ensure your VMC environment has the following prerequisites:

- Create a non-exclusive, DHCP-enabled, NSX-T network segment and subnet. Other virtual machines (VMs) can be hosted on the subnet, but at least eight IP addresses must be available for the OpenShift Container Platform deployment.
- Allocate two IP addresses, outside the DHCP range, and configure them with reverse DNS records.
    - A DNS record for **api.<cluster_name>.<base_domain>** pointing to the allocated IP address.

  - A DNS record for ***.apps.<cluster_name>.<base_domain>** pointing to the allocated IP address.

- Configure the following firewall rules:
  - An ANY:ANY firewall rule between the OpenShift Container Platform compute network and the internet. This is used by nodes and applications to download container images.
  - An ANY:ANY firewall rule between the installation host and the software-defined data center (SDDC) management network on port 443. This allows you to upload the Red Hat Enterprise Linux CoreOS (RHCOS) OVA during deployment.
  - An HTTPS firewall rule between the OpenShift Container Platform compute network and vCenter. This connection allows OpenShift Container Platform to communicate with vCenter for provisioning and managing nodes, persistent volume claims (PVCs), and other resources.
- You must have the following information to deploy OpenShift Container Platform:
  - The OpenShift Container Platform cluster name, such as **vmc-prod-1**.
  - The base DNS name, such as **companyname.com**.
  - If not using the default, the pod network CIDR and services network CIDR must be identified, which are set by default to **10.128.0.0/14** and **172.30.0.0/16**, respectively. These CIDRs are used for pod-to-pod and pod-to-service communication and are not accessible externally; however, they must not overlap with existing subnets in your organization.
  - The following vCenter information:
    - vCenter hostname, username, and password
    - Datacenter name, such as **SDDC-Datacenter**
    - Cluster name, such as **Cluster-1**
    - Network name
    - Datastore name, such as **WorkloadDatastore**

    NOTE

    It is recommended to move your vSphere cluster to the VMC **Compute-ResourcePool** resource pool after your cluster installation is finished.

- A Linux-based host deployed to VMC as a bastion.
  - The bastion host can be Red Hat Enterprise Linux (RHEL) or any another Linux-based host; it must have internet connectivity and the ability to upload an OVA to the ESXi hosts.
  - Download and install the OpenShift CLI tools to the bastion host.
    - The **openshift-install** installation program
    - The OpenShift CLI (**oc**) tool

NOTE

You cannot use the VMware NSX Container Plugin for Kubernetes (NCP), and NSX is not used as the OpenShift SDN. The version of NSX currently available with VMC is incompatible with the version of NCP certified with OpenShift Container Platform.

However, the NSX DHCP service is used for virtual machine IP management with the full-stack automated OpenShift Container Platform deployment and with nodes provisioned, either manually or automatically, by the Machine API integration with vSphere. Additionally, NSX firewall rules are created to enable access with the OpenShift Container Platform cluster and between the bastion host and the VMC vSphere hosts.

4.1.1. VMC Sizer tool

VMware Cloud on AWS is built on top of AWS bare metal infrastructure; this is the same bare metal infrastructure which runs AWS native services. When a VMware cloud on AWS software-defined data center (SDDC) is deployed, you consume these physical server nodes and run the VMware ESXi hypervisor in a single tenant fashion. This means the physical infrastructure is not accessible to anyone else using VMC. It is important to consider how many physical hosts you will need to host your virtual infrastructure.

To determine this, VMware provides the VMC on AWS Sizer. With this tool, you can define the resources you intend to host on VMC:

- Types of workloads
- Total number of virtual machines
- Specification information such as:
  - Storage requirements
  - vCPUs
  - vRAM
  - Overcommit ratios

With these details, the sizer tool can generate a report, based on VMware best practices, and recommend your cluster configuration and the number of hosts you will need.

4.2. VSPHERE PREREQUISITES

- 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 block registry storage. For more information on persistent storage, see Understanding persistent storage.
- 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.

4.3. INTERNET ACCESS FOR OPENSHIFT CONTAINER PLATFORM

In OpenShift Container Platform 4.12, you require access to the internet to install your cluster.

You must have internet access to:

- Access OpenShift Cluster Manager Hybrid Cloud Console 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.

4.4. 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 7.0 Update 2 or later
- Version 8.0 Update 1 or later

You can host the VMware vSphere infrastructure on-premise or on a VMware Cloud Verified provider that meets the requirements outlined in the following table:

Table 4.1. Version requirements for vSphere virtual environments

Virtual environment product	Required version
VMware virtual hardware	15 or later
vSphere ESXi hosts	7.0 Update 2 or later; 8.0 Update 1 or later
vCenter host	7.0 Update 2 or later; 8.0 Update 1 or later

IMPORTANT

Installing a cluster on VMware vSphere versions 7.0 and 7.0 Update 1 is deprecated. These versions are still fully supported, but all vSphere 6.x versions are no longer supported. Version 4.12 of OpenShift Container Platform requires VMware virtual hardware version 15 or later. To update the hardware version for your vSphere virtual machines, see the "Updating hardware on nodes running in vSphere" article in the *Updating clusters* section.

Table 4.2. Minimum supported vSphere version for VMware components

Component	Minimum supported versions	Description
Hypervisor	vSphere 7.0 Update 2 (or later) or vSphere 8.0 Update 1 (or later) with virtual hardware version 15	This 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.
Storage with in-tree drivers	vSphere 7.0 Update 2 or later; 8.0 Update 1 or later	This plugin creates vSphere storage by using the in-tree storage drivers for vSphere included in OpenShift Container Platform.

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.

4.5. 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 4.3. 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

Protocol	Port	Description
	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
	10256	openshift-sdn
UDP	**4789**	virtual extensible LAN (VXLAN)
	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	IPsec Encapsulating Security Payload (ESP)

Table 4.4. Ports used for all-machine to control plane communications

Protocol	Port	Description
TCP	**6443**	Kubernetes API

Table 4.5. Ports used for control plane machine to control plane machine communications

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

4.6. VMWARE VSPHERE CSI DRIVER OPERATOR REQUIREMENTS

To install the vSphere CSI Driver Operator, the following requirements must be met:

- VMware vSphere version: 7.0 Update 2 or later; 8.0 Update 1 or later
- vCenter version: 7.0 Update 2 or later; 8.0 Update 1 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.

Additional resources

- To remove a third-party 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.

4.7. VCENTER REQUIREMENTS

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

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. While most of the privileges are always required, some 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 amend vSphere roles for the specified objects to grant the required privileges.

An additional role is required if the installation program is to create a vSphere virtual machine folder.

Example 4.1. Roles and privileges required for installation in vSphere API

vSphere object for role	When required	Required privileges 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 will be created in the cluster root	**Host.Config.Storage Resource.AssignVMToPool VApp.AssignResourcePool VApp.Import VirtualMachine.Config.AddNewDisk**
vSphere vCenter Resource Pool	If an existing resource pool is provided	**Host.Config.Storage 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.Add**

vSphere object for role	When required	Required privileges in vSphere API
		RemoveDevice **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** **VirtualMachine.Config.ResetGuestInfo** **VirtualMachine.Config.Resource** **VirtualMachine.Config.Settings** **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 Datacenter	If the installation program creates the virtual machine folder. For UPI, **VirtualMachine.Inventory.Create** and **VirtualMachine.Inventory.Delete** privileges are optional if your cluster does not use the Machine API.	**InventoryService.Tagging.ObjectAttachable** **Resource.AssignVMToPool** **VApp.Import** **VirtualMachine.Config.AddExistingDisk** **VirtualMachine.Config.AddNewDisk**

vSphere object for role	When required	Required privileges in vSphere API
		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 VirtualMachine.Config.ResetGuestInfo VirtualMachine.Config.Resource VirtualMachine.Config.Settings 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.DeployTemplate VirtualMachine.Provisioning.MarkAsTemplate Folder.Create Folder.Delete

Example 4.2. Roles and privileges required for installation in vCenter graphical user interface (GUI)

vSphere object for role	When required	Required privileges in vCenter GUI
vSphere 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" "Profile-driven storage"."Profile-driven storage update" "Profile-driven storage"."Profile-driven storage view"**
vSphere vCenter Cluster	If VMs will be created 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 an existing resource pool is provided	**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 object for role	When required	Required privileges in vCenter GUI
vSphere Datastore	Always	**Datastore."Allocate space" Datastore."Browse datastore" Datastore."Low level file operations" "vSphere Tagging"."Assign or Unassign vSphere Tag on Object"**
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"**

vSphere object for role	When required	Required privileges in vCenter GUI
		"Virtual machine"."Change Configuration".Rename **"Virtual machine"."Change Configuration"."Reset guest information"** **"Virtual machine"."Change Configuration"."Change resource"** **"Virtual machine"."Change Configuration"."Change Settings"** **"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 Datacenter	If the installation program creates the virtual machine folder. For UPI, **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"**

vSphere object for role	When required	Required privileges in vCenter GUI
		"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 Settings" "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"

vSphere object for role	When required	Required privileges in vCenter GUI
		"Virtual machine"."Edit Inventory"."Create from existing" **"Virtual machine"."Edit Inventory"."Remove"** **"Virtual machine".Provisioning."Clone virtual machine"** **"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 propogate the permissions to child objects. These settings vary depending on whether or not you install the cluster into an existing folder.

Example 4.3. Required permissions and propagation settings

vSphere object	When required	Propagate to children	Permissions required
vSphere vCenter	Always	False	Listed required privileges
vSphere vCenter Datacenter	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 object	When required	Propagate to children	Permissions required
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.

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.

- OpenShift Container Platform generally supports compute-only vMotion, 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.

- Using Storage vMotion can cause issues and is not supported. If you are using 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 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.

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 is destroyed during the cluster installation process. A minimum of 800 GB of storage is required to use a standard cluster.

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

Cluster limits

Available resources vary between clusters. The number of possible clusters within a vCenter is limited 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.

Networking requirements

You must use the Dynamic Host Configuration Protocol (DHCP) for the network and ensure that the DHCP server is configured to provide persistent IP addresses to the cluster machines. In the DHCP lease, you must configure the DHCP to use the default gateway. All nodes must be in the same VLAN. You cannot scale the cluster using a second VLAN as a Day 2 operation. Additionally, you must create the following networking resources before you install the OpenShift Container Platform cluster:

> **NOTE**
>
> It is recommended that each OpenShift Container Platform node in the cluster must have access to a Network Time Protocol (NTP) server that is discoverable via DHCP. Installation is possible without an NTP server. However, asynchronous server clocks will cause errors, which NTP server prevents.

Required IP Addresses

An installer-provisioned vSphere installation requires two static IP addresses:

- The **API** address is used to access the cluster API.
- The **Ingress** address is used for cluster ingress traffic.

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

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 4.6. Required DNS records

Component	Record	Description
API VIP	**api.<cluster_name>.<base_domain>.**	This DNS A/AAAA or CNAME 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.

Component	Record	Description
Ingress VIP	***.apps.<cluster_name>.<base_domain>.**	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.

4.8. GENERATING A KEY PAIR FOR CLUSTER NODE SSH ACCESS

During an OpenShift Container Platform installation, you can provide an SSH public key to the installation program. The key is passed to the Red Hat Enterprise Linux CoreOS (RHCOS) nodes through their Ignition config files and is used to authenticate SSH access to the nodes. The key is added to the **~/.ssh/authorized_keys** list for the **core** user on each node, which enables password-less authentication.

After the key is passed to the nodes, 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 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 such as AWS key pairs.

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> 1
```

1 Specify 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 FIPS validated or Modules In Process cryptographic libraries on 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> (1)
```

(1) Specify 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.

4.9. 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 4.12 release notes* document.

Procedure

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

2. Select your infrastructure provider.

3. Navigate to the page for your installation type, download the installation program that corresponds with your host operating system and architecture, and place the file in the directory where you will store the installation configuration files.

   > **IMPORTANT**
   >
   > The installation program creates several files on the computer that you use to install your cluster. You must keep the installation program and the files that the installation program creates after you finish installing the cluster. Both files are required to delete the cluster.

   > **IMPORTANT**
   >
   > 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.

4. 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
   ```

5. Download your installation pull secret from the 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.

4.10. 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

1. 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.

2. 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
```

4.11. VMWARE VSPHERE REGION AND ZONE ENABLEMENT

You can deploy an OpenShift Container Platform cluster to multiple vSphere datacenters that run in a single VMware vCenter. Each datacenter 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

VMware vSphere region and zone enablement 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 .

The default installation configuration deploys a cluster to a single vSphere datacenter. If you want to deploy a cluster to multiple vSphere datacenters, 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 datacenters 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 datacenter.

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

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

Table 4.7. Example of a configuration with multiple vSphere datacenters that run in a single VMware vCenter

Datacenter (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

4.12. CREATING THE INSTALLATION CONFIGURATION FILE

You can customize the OpenShift Container Platform cluster you install on VMware vSphere.

Prerequisites

- Obtain the OpenShift Container Platform installation program and the pull secret for your cluster.
- Obtain service principal permissions at the subscription level.

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> 1
    ```

    1 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.

vi. Select the default vCenter datastore to use.

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.

xiii. Paste the pull secret from the Red Hat OpenShift Cluster Manager .

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.

4.12.1. Installation configuration parameters

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

NOTE

After installation, you cannot modify these parameters in the **install-config.yaml** file.

4.12.1.1. Required configuration parameters

Required installation configuration parameters are described in the following table:

Table 4.8. Required parameters

Parameter	Description	Values
apiVersion	The API version for the **install-config.yaml** content. The current version is **v1**. The installation program may also support older API versions.	String
baseDomain	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.	A fully-qualified domain or subdomain name, such as **example.com**.
metadata	Kubernetes resource **ObjectMeta**, from which only the **name** parameter is consumed.	Object
metadata.name	The name of the cluster. DNS records for the cluster are all subdomains of **{{.metadata.name}}.{{.baseDomain}}**.	String of lowercase letters and hyphens (**-**), such as **dev**.

Parameter	Description	Values
platform	The configuration for the specific platform upon which to perform the installation: **alibabacloud**, **aws**, **baremetal**, **azure**, **gcp**, **ibmcloud**, **nutanix**, **openstack**, **ovirt**, **vsphere**, or **{}**. For additional information about **platform.<platform>** parameters, consult the table for your specific platform that follows.	Object
pullSecret	Get a pull secret from the Red Hat OpenShift Cluster Manager to authenticate downloading container images for OpenShift Container Platform components from services such as Quay.io.	`{ "auths":{ "cloud.openshift.com":{ "auth":"b3Blb=", "email":"you@example.com" }, "quay.io":{ "auth":"b3Blb=", "email":"you@example.com" } } }`

4.12.1.2. 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 provide different IP address blocks than the defaults.

Only IPv4 addresses are supported.

NOTE

Globalnet is not supported with Red Hat OpenShift Data Foundation disaster recovery solutions. For regional disaster recovery scenarios, ensure that you use a nonoverlapping range of private IP addresses for the cluster and service networks in each cluster.

Table 4.9. Network parameters

Parameter	Description	Values

Parameter	Description	Values
networking	The configuration for the cluster network.	Object NOTE You cannot modify parameters specified by the **networking** object after installation.
networking.networkType	The Red Hat OpenShift Networking network plugin to install.	Either **OpenShiftSDN** or **OVNKubernetes**. **OpenShiftSDN** is a CNI plugin for all-Linux networks. **OVNKubernetes** is a CNI plugin for Linux networks and hybrid networks that contain both Linux and Windows servers. The default value is **OVNKubernetes**.

networking.clusterNetwork	The IP address blocks for pods.
	
	The default value is **10.128.0.0/14** with a host prefix of **/23**.
	
	If you specify multiple IP address blocks, the blocks must not overlap.
		An array of objects. For example:
		
		`networking:`
		`  clusterNetwork:`
		`  - cidr: 10.128.0.0/14`
		`    hostPrefix: 23`

networking.clusterNetwork.cidr	Required if you use **networking.clusterNetwork**. An IP address block. An IPv4 network.	An IP address block in Classless Inter-Domain Routing (CIDR) notation. The prefix length for an IPv4 block is between **0** and **32**.
networking.clusterNetwork.hostPrefix	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.	A subnet prefix. The default value is **23**.

networking.serviceNetwork	The IP address block for services. The default value is **172.30.0.0/16**.
	
	The OpenShift SDN and OVN-Kubernetes network plugins support only a single IP address block for the service network.
		An array with an IP address block in CIDR format. For example:
		
		`networking:`
		`  serviceNetwork:`
		`   - 172.30.0.0/16`

Parameter	Description	Values

networking.machineNetwork	The IP address blocks for machines.
	
	If you specify multiple IP address blocks, the blocks must not overlap.
		An array of objects. For example:
		
		`networking:`
		`  machineNetwork:`
		`  - cidr: 10.0.0.0/16`

networking.machineNetwork.cidr	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. For libvirt, the default value is **192.168.126.0/24**.	An IP network block in CIDR notation. For example, **10.0.0.0/16**. NOTE Set the **networking.machineNetwork** to match the CIDR that the preferred NIC resides in.

4.12.1.3. Optional configuration parameters

Optional installation configuration parameters are described in the following table:

Table 4.10. Optional parameters

Parameter	Description	Values
additionalTrustBundle	A PEM-encoded X.509 certificate bundle that is added to the nodes' trusted certificate store. This trust bundle may also be used when a proxy has been configured.	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 *Installing*.	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**.	String

Parameter	Description	Values
capabilities.additionalEnabledCapabilities	Extends the set of optional capabilities beyond what you specify in **baselineCapabilitySet**. You may specify multiple capabilities in this parameter.	String array
compute	The configuration for the machines that comprise the compute nodes.	Array of **MachinePool** objects.
compute.architecture	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).	String
compute.hyperthreading	Whether to enable or disable simultaneous multithreading, or **hyperthreading**, on compute machines. By default, simultaneous multithreading is enabled to increase the performance of your machines' cores. IMPORTANT If you disable simultaneous multithreading, ensure that your capacity planning accounts for the dramatically decreased machine performance.	**Enabled** or **Disabled**
compute.name	Required if you use **compute**. The name of the machine pool.	**worker**
compute.platform	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.	**alibabacloud**, **aws**, **azure**, **gcp**, **ibmcloud**, **nutanix**, **openstack**, **ovirt**, **vsphere**, or **{}**
compute.replicas	The number of compute machines, which are also known as worker machines, to provision.	A positive integer greater than or equal to **2**. The default value is **3**.

Parameter	Description	Values
featureSet	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".	String. The name of the feature set to enable, such as **TechPreviewNoUpgrade**.
controlPlane	The configuration for the machines that comprise the control plane.	Array of **MachinePool** objects.
controlPlane.architecture	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).	String
controlPlane.hyperthreading	Whether to enable or disable simultaneous multithreading, or **hyperthreading**, on control plane machines. By default, simultaneous multithreading is enabled to increase the performance of your machines' cores. IMPORTANT If you disable simultaneous multithreading, ensure that your capacity planning accounts for the dramatically decreased machine performance.	**Enabled** or **Disabled**
controlPlane.name	Required if you use **controlPlane**. The name of the machine pool.	**master**
controlPlane.platform	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.	**alibabacloud**, **aws**, **azure**, **gcp**, **ibmcloud**, **nutanix**, **openstack**, **ovirt**, **vsphere**, or **{}**
controlPlane.replicas	The number of control plane machines to provision.	The only supported value is **3**, which is the default value.

Parameter	Description	Values
credentialsMode	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. NOTE Not all CCO modes are supported for all cloud providers. For more information about CCO modes, see the *Cloud Credential Operator* entry in the *Cluster Operators reference* content. NOTE If your AWS account has service control policies (SCP) enabled, you must configure the **credentialsMode** parameter to **Mint**, **Passthrough** or **Manual**.	**Mint**, **Passthrough**, **Manual** or an empty string (**""**).

Parameter	Description	Values
fips	Enable or disable FIPS mode. The default is **false** (disabled). 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 [Installing the system in FIPS mode]. The use of FIPS validated or Modules In Process cryptographic libraries is only supported on OpenShift Container Platform deployments on the **x86_64**, **ppc64le**, and **s390x** architectures. NOTE If you are using Azure File storage, you cannot enable FIPS mode.	**false** or **true**
imageContentSources	Sources and repositories for the release-image content.	Array of objects. Includes a **source** and, optionally, **mirrors**, as described in the following rows of this table.
imageContentSources.source	Required if you use **imageContentSources**. Specify the repository that users refer to, for example, in image pull specifications.	String

Parameter	Description	Values
imageContentSources.mirrors	Specify one or more repositories that may also contain the same images.	Array of strings
publish	How to publish or expose the user-facing endpoints of your cluster, such as the Kubernetes API, OpenShift routes.	**Internal** or **External**. The default value is **External**. Setting this field to **Internal** is not supported on non-cloud platforms. IMPORTANT: If the value of the field is set to **Internal**, the cluster will become non-functional. For more information, refer to BZ#1953035.
sshKey	The SSH key to authenticate access to 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.	For example, **sshKey: ssh-ed25519 AAAA..**.

4.12.1.4. Additional VMware vSphere configuration parameters

Additional VMware vSphere configuration parameters are described in the following table.

NOTE

The **platform.vsphere** parameter prefixes each parameter listed in the table.

Table 4.11. Additional VMware vSphere cluster parameters

Parameter	Description	Values
vCenter	The fully-qualified hostname or IP address of the vCenter server.	String

Parameter	Description	Values
username	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.	String
password	The password for the vCenter user name.	String
datacenter	The name of the data center to use in the vCenter instance.	String
defaultDatastore	The name of the default datastore to use for provisioning volumes.	String
folder	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.	String, for example, **/<datacenter_name>/vm/<folder_name>/<subfolder_name>**.
resourcePool	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 **/<datacenter_name>/host/<cluster_name>/Resources**.	String, for example, **/<datacenter_name>/host/<cluster_name>/Resources/<resource_pool_name>/<optional_nested_resource_pool_name>**.
network	The network in the vCenter instance that contains the virtual IP addresses and DNS records that you configured.	String
cluster	The vCenter cluster to install the OpenShift Container Platform cluster in.	String
apiVIPs	The virtual IP (VIP) address that you configured for control plane API access. NOTE 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.	An IP address, for example **128.0.0.1**.

Parameter	Description	Values
ingressVIPs	The virtual IP (VIP) address that you configured for cluster ingress. NOTE 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.	An IP address, for example **128.0.0.1**.
diskType	Optional. The disk provisioning method. This value defaults to the vSphere default storage policy if not set.	Valid values are **thin**, **thick**, or **eagerZeroedThick**.

4.12.1.5. Optional VMware vSphere machine pool configuration parameters

Optional VMware vSphere machine pool configuration parameters are described in the following table.

NOTE

The **platform.vsphere** parameter prefixes each parameter listed in the table.

Table 4.12. Optional VMware vSphere machine pool parameters

Parameter	Description	Values
clusterOSImage	The location from which the installation program downloads the RHCOS image. You must set this parameter to perform an installation in a restricted network.	An HTTP or HTTPS URL, optionally with a SHA-256 checksum. For example, **https://mirror.openshift.com/images/rhcos-<version>-vmware.<architecture>.ova**.
osDisk.diskSizeGB	The size of the disk in gigabytes.	Integer
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.	Integer

Parameter	Description	Values
coresPerSocket	The number of cores per socket in a virtual machine. The number of virtual sockets on the virtual machine is **platform.vsphere.cpus**/**platform.vsphere.coresPerSocket**. The default value for control plane nodes and worker nodes is **4** and **2**, respectively.	Integer
memoryMB	The size of a virtual machine's memory in megabytes.	Integer

4.12.1.6. Region and zone enablement configuration parameters

To use the region and zone enablement feature, you must specify region and zone enablement parameters in your installation file.

IMPORTANT

Before you modify the **install-config.yaml** file to configure a region and zone enablement environment, read the "VMware vSphere region and zone enablement" and the "Configuring regions and zones for a VMware vCenter" sections.

NOTE

The **platform.vsphere** parameter prefixes each parameter listed in the table.

Table 4.13. Region and zone enablement parameters

Parameter	Description	Values
failureDomains	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.	String
failureDomains.name	The name of the failure domain. The machine pools use this name to reference the failure domain.	String
failureDomains.server	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.	String
failureDomains.region	You define a region by using a tag from the **openshift-region** tag category. The tag must be attached to the vCenter datacenter.	String

Parameter	Description	Values
failureDomains.zone	You define a zone by using a tag from the **openshift-zone** tag category. The tag must be attached to the vCenter datacenter.	String
failureDomains.topology.computeCluster	This parameter defines the compute cluster associated with the failure domain. If you do not define this parameter in your configuration, the compute cluster takes the value of **platform.vsphere.cluster** and **platform.vsphere.datacenter**.	String
failureDomains.topology.folder	The absolute path of an existing folder where the installation program creates the virtual machines. If you do not define this parameter in your configuration, the folder takes the value of **platform.vsphere.folder**.	String
failureDomains.topology.datacenter	Defines the datacenter where OpenShift Container Platform virtual machines (VMs) operate. If you do not define this parameter in your configuration, the datacenter defaults to **platform.vsphere.datacenter**.	String
failureDomains.topology.datastore	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.	String
failureDomains.topology.networks	Lists any network in the vCenter instance that contains the virtual IP addresses and DNS records that you configured. If you do not define this parameter in your configuration, the network takes the value of **platform.vsphere.network**.	String
failureDomains.topology.resourcePool	Optional: The absolute path of an existing resource pool where the installation program creates the virtual machines, for example, **/<datacenter_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 **/<datacenter_name>/host/<cluster_name>/Resources**.	String

4.12.2. 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 modify the values of the required parameters.

```
apiVersion: v1
baseDomain: example.com 1
compute: 2
  name: worker
  replicas: 3
  platform:
    vsphere: 3
      cpus: 2
      coresPerSocket: 2
      memoryMB: 8192
      osDisk:
        diskSizeGB: 120
controlPlane: 4
  name: master
  replicas: 3
  platform:
    vsphere: 5
      cpus: 4
      coresPerSocket: 2
      memoryMB: 16384
      osDisk:
        diskSizeGB: 120
metadata:
  name: cluster 6
networking:
  clusterNetwork:
  - cidr: 10.128.0.0/14
    hostPrefix: 23
  machineNetwork:
  - cidr: 10.0.0.0/16
  networkType: OVNKubernetes 7
  serviceNetwork:
  - 172.30.0.0/16
platform:
  vsphere:
    vcenter: your.vcenter.server
    username: username
    password: password
    datacenter: datacenter
    defaultDatastore: datastore
    folder: folder
    resourcePool: resource_pool 8
    diskType: thin 9
    network: VM_Network
    cluster: vsphere_cluster_name 10
    apiVIPs:
      - api_vip
    ingressVIPs:
      - ingress_vip
```

```
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 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. Only one control plane pool is used.

3 5 Optional: Provide additional configuration for the machine pool parameters for the compute and control plane machines.

6 The cluster name that you specified in your DNS records.

8 Optional: Provide 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 The vSphere disk provisioning method.

10 The vSphere cluster to install the OpenShift Container Platform cluster in.

7 The cluster network plugin to install. The supported values are **OVNKubernetes** and **OpenShiftSDN**. The default value is **OVNKubernetes**.

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.

4.12.3. Configuring the cluster-wide proxy during installation

Production environments can deny direct access to the internet and instead have an HTTP or HTTPS proxy available. You can configure a new OpenShift Container Platform cluster to use a proxy by configuring the proxy settings in the **install-config.yaml** file.

Prerequisites

- You have an existing **install-config.yaml** file.
- You 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 Platform (GCP), 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> 1
  httpsProxy: https://<username>:<pswd>@<ip>:<port> 2
  noProxy: example.com 3
additionalTrustBundle: | 4
    -----BEGIN CERTIFICATE-----
    <MY_TRUSTED_CA_CERT>
    -----END CERTIFICATE-----
additionalTrustBundlePolicy: <policy_to_add_additionalTrustBundle> 5
```

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

2 A proxy URL to use for creating HTTPS connections outside the cluster.

3 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.

4 If provided, the installation program generates a config map that is named **user-ca-bundle** in the **openshift-config** namespace that contains one or more additional CA certificates that are required for proxying HTTPS connections. The Cluster Network Operator then creates a **trusted-ca-bundle** config map that merges these contents with the Red Hat Enterprise Linux CoreOS (RHCOS) trust bundle, and this config map is referenced in the **trustedCA** field of the **Proxy** object. The **additionalTrustBundle** field is required unless the proxy's identity certificate is signed by an authority from the RHCOS trust bundle.

5 Optional: The policy to determine 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**.

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.

4.12.4. Configuring regions and zones for a VMware vCenter

You can modify the default installation configuration file to deploy an OpenShift Container Platform cluster to multiple vSphere datacenters that run in a single VMware vCenter.

IMPORTANT

VMware vSphere region and zone enablement 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 .

IMPORTANT

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

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 datacenter objects for your VMware vCenter server. Consider specifying multiple failure domains if you need to provision virtual machine nodes in different datacenters, clusters, datastores, and other components. To enable regions and zones, you must define multiple failure domains for your OpenShift Container Platform cluster.

NOTE

You cannot change a failure domain after you installed an OpenShift Container Platform cluster on the VMware vSphere platform. You can add additional failure domains after cluster installation.

Procedure

1. Enter the following **govc** command-line tool commands to create the **openshift-region** and **openshift-zone** vCenter tag categories:

   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. To create a region tag for each region vSphere datacenter where you want to deploy your cluster, enter the following command in your terminal:

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

3. To create a zone tag for each vSphere cluster where you want to deploy your cluster, enter the following command:

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

4. Attach region tags to each vCenter datacenter object by entering the following command:

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

5. Attach the zone tags to each vCenter datacenter object by entering the following command:

```
$ govc tags.attach -c <zone_tag_category> <zone_tag_1> /<datacenter_1>/host/vcs-mdcnc-workload-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 datacenters defined in a vSphere center

```
apiVersion: v1
baseDomain: example.com
featureSet: TechPreviewNoUpgrade 1
compute:
  name: worker
  replicas: 3
  vsphere:
    zones: 2
      - "<machine_pool_zone_1>"
      - "<machine_pool_zone_2>"
controlPlane:
  name: master
  replicas: 3
  vsphere:
    zones: 3
      - "<machine_pool_zone_1>"
      - "<machine_pool_zone_2>"
metadata:
  name: cluster
platform:
  vsphere:
    vcenter: <vcenter_server> 4
    username: <username> 5
    password: <password> 6
    datacenter: datacenter 7
    defaultDatastore: datastore 8
    folder: "/<datacenter_name>/vm/<folder_name>/<subfolder_name>" 9
    cluster: cluster 10
    resourcePool: "/<datacenter_name>/host/<cluster_name>/Resources/<resource_pool_name>" 11
    diskType: thin
    failureDomains: 12
    - name: <machine_pool_zone_1> 13
      region: <region_tag_1> 14
      zone: <zone_tag_1> 15
      topology: 16
        datacenter: <datacenter1> 17
        computeCluster: "/<datacenter1>/host/<cluster1>" 18
        resourcePool: "/<datacenter1>/host/<cluster1>/Resources/<resourcePool1>" 19
        networks: 20
        - <VM_Network1_name>
        datastore: "/<datacenter1>/datastore/<datastore1>" 21
    - name: <machine_pool_zone_2>
      region: <region_tag_2>
      zone: <zone_tag_2>
      topology:
        datacenter: <datacenter2>
        computeCluster: "/<datacenter2>/host/<cluster2>"
        networks:
        - <VM_Network2_name>
        datastore: "/<datacenter2>/datastore/<datastore2>"
```

```
      resourcePool: "/<datacenter2>/host/<cluster2>/Resources/<resourcePool2>"
      folder: "/<datacenter2>/vm/<folder2>"
# ...
```

1 You must define set the **TechPreviewNoUpgrade** as the value for this parameter, so that you can use the VMware vSphere region and zone enablement feature.

2 3 An optional parameter for specifying a vCenter cluster. You define a zone by using a tag from the **openshift-zone** tag category. If you do not define this parameter, nodes will be distributed among all defined failure-domains.

4 5 6 7 8 9 10 11 The default vCenter topology. The installation program uses this topology information to deploy the bootstrap node. Additionally, the topology defines the default datastore for vSphere persistent volumes.

12 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. If you do not define this parameter, the installation program uses the default vCenter topology.

13 Defines the name of the failure domain. Each failure domain is referenced in the **zones** parameter to scope a machine pool to the failure domain.

14 You define a region by using a tag from the **openshift-region** tag category. The tag must be attached to the vCenter datacenter.

15 You define a zone by using a tag from the **openshift-zone tag** category. The tag must be attached to the vCenter datacenter.

16 Specifies the vCenter resources associated with the failure domain.

17 An optional parameter for defining the vSphere datacenter that is associated with a failure domain. If you do not define this parameter, the installation program uses the default vCenter topology.

18 An optional parameter for stating the absolute file path for the compute cluster that is associated with the failure domain. If you do not define this parameter, the installation program uses the default vCenter topology.

19 An optional parameter for the installer-provisioned infrastructure. The parameter sets the absolute path of an existing resource pool where the installation program creates the virtual machines, for example, **/<datacenter_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_datacenter/host/example_cluster/Resources**.

20 An optional parameter that lists any network in the vCenter instance that contains the virtual IP addresses and DNS records that you configured. If you do not define this parameter, the installation program uses the default vCenter topology.

21 An optional parameter for specifying a datastore to use for provisioning volumes. If you do not define this parameter, the installation program uses the default vCenter topology.

4.13. 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**
  For more information on these fields, refer to *Installation configuration parameters*.

> NOTE
>
> Set the **networking.machineNetwork** to match the CIDR that the preferred NIC resides in.

> IMPORTANT
>
> The CIDR range **172.17.0.0/16** is reserved by libVirt. You cannot use this range or any range that overlaps with this range for any 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 advanced network configuration.

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

4.14. 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.yml** 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 examples:

   Specify a different VXLAN port for the OpenShift SDN network provider

```
apiVersion: operator.openshift.io/v1
kind: Network
metadata:
  name: cluster
spec:
  defaultNetwork:
    openshiftSDNConfig:
      vxlanPort: 4800
```

   Enable IPsec for the OVN-Kubernetes network provider

```
apiVersion: operator.openshift.io/v1
kind: Network
metadata:
  name: cluster
spec:
  defaultNetwork:
    ovnKubernetesConfig:
      ipsecConfig: {}
```

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.

4.15. CLUSTER NETWORK OPERATOR CONFIGURATION

The configuration for the cluster network is specified as part of the Cluster Network Operator (CNO) configuration and stored in a custom resource (CR) object that is named **cluster**. The CR specifies the fields for the **Network** API in the **operator.openshift.io** API group.

The CNO configuration inherits the following fields during cluster installation from the **Network** API in the **Network.config.openshift.io** API group and these fields cannot be changed:

clusterNetwork

IP address pools from which pod IP addresses are allocated.

serviceNetwork

IP address pool for services.

defaultNetwork.type

Cluster network plugin, such as OpenShift SDN or OVN-Kubernetes.

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**.

4.15.1. Cluster Network Operator configuration object

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

Table 4.14. 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**	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:
		`spec:`
		`  clusterNetwork:`
		`  - cidr: 10.128.0.0/19`
		`    hostPrefix: 23`
		`  - cidr: 10.128.32.0/19`
		`    hostPrefix: 23`
		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.

spec.serviceNetwork	**array**	A block of IP addresses for services. The OpenShift SDN and OVN-Kubernetes network plugins support only a single IP address block for the service network. For example:
		`spec:`
		`  serviceNetwork:`
		`  - 172.30.0.0/14`
		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.

spec.defaultNetwork	**object**	Configures the network plugin for the cluster network.

Field	Type	Description
spec.kubeProxyConfig	**object**	The fields for this object specify the kube-proxy configuration. If you are using the OVN-Kubernetes cluster network plugin, the kube-proxy configuration has no effect.

defaultNetwork object configuration

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

Table 4.15. defaultNetwork object

Field	Type	Description
type	**string**	Either **OpenShiftSDN** or **OVNKubernetes**. The Red Hat OpenShift Networking network plugin is selected during installation. This value cannot be changed after cluster installation. **NOTE** OpenShift Container Platform uses the OVN-Kubernetes network plugin by default.
openshiftSDNConfig	**object**	This object is only valid for the OpenShift SDN network plugin.
ovnKubernetesConfig	**object**	This object is only valid for the OVN-Kubernetes network plugin.

Configuration for the OpenShift SDN network plugin

The following table describes the configuration fields for the OpenShift SDN network plugin:

Table 4.16. openshiftSDNConfig object

Field	Type	Description
mode	**string**	Configures the network isolation mode for OpenShift SDN. The default value is **NetworkPolicy**. The values **Multitenant** and **Subnet** are available for backwards compatibility with OpenShift Container Platform 3.x but are not recommended. This value cannot be changed after cluster installation.

Field	Type	Description
mtu	**integer**	The maximum transmission unit (MTU) for the VXLAN 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. 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. If your cluster requires different MTU values for different nodes, you must set this value to **50** 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 **1450**. This value cannot be changed after cluster installation.
vxlanPort	**integer**	The port to use for all VXLAN packets. The default value is **4789**. This value cannot be changed after cluster installation. If you are running in a virtualized environment with existing nodes that are part of another VXLAN network, then you might be required to change this. For example, when running an OpenShift SDN overlay on top of VMware NSX-T, you must select an alternate port for the VXLAN, because both SDNs use the same default VXLAN port number. On Amazon Web Services (AWS), you can select an alternate port for the VXLAN between port **9000** and port **9999**.

Example OpenShift SDN configuration

```
defaultNetwork:
  type: OpenShiftSDN
  openshiftSDNConfig:
    mode: NetworkPolicy
    mtu: 1450
    vxlanPort: 4789
```

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

Table 4.17. ovnKubernetesConfig object

Field	Type	Description

Field	Type	Description
mtu	**integer**	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. 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. 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**.
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 an empty object to enable IPsec encryption.
policyAuditConfig	**object**	Specify a configuration object for customizing network policy audit logging. If unset, the defaults audit log settings are used.
gatewayConfig	**object**	Optional: Specify a configuration object for customizing how egress traffic is sent to the node gateway. NOTE 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.

Field	Type	Description
v4InternalSubnet	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**. This field cannot be changed after installation.	The default value is **100.64.0.0/16**.

Field	Type	Description
v6InternalSubnet	If your existing network infrastructure overlaps with the **fd98::/48** 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. This field cannot be changed after installation.	The default value is **fd98::/48**.

Table 4.18. **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
destination	string	One of the following additional audit log targets: **libc** The libc **syslog()** function of the journald process on the host. **udp:<host>:<port>** A syslog server. Replace **<host>:<port>** with the host and port of the syslog server. **unix:<file>** A Unix Domain Socket file specified by **<file>**. **null** Do not send the audit logs to any additional target.
syslogFacility	string	The syslog facility, such as **kern**, as defined by RFC5424. The default value is **local0**.

Table 4.19. **gatewayConfig** object

Field	Type	Description
routingViaHost	**boolean**	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**. 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.

Example OVN-Kubernetes configuration with IPSec enabled

```
defaultNetwork:
  type: OVNKubernetes
  ovnKubernetesConfig:
    mtu: 1400
    genevePort: 6081
    ipsecConfig: {}
```

kubeProxyConfig object configuration

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

Table 4.20. **kubeProxyConfig** object

Field	Type	Description
iptablesSyncPeriod	**string**	The refresh period for **iptables** rules. The default value is **30s**. Valid suffixes include **s**, **m**, and **h** and are described in the Go **time** package documentation. NOTE Because of performance improvements introduced in OpenShift Container Platform 4.3 and greater, adjusting the **iptablesSyncPeriod** parameter is no longer necessary.

proxyArguments.iptables-min-sync-period	**array**	The minimum duration before refreshing **iptables** rules. This field ensures that the refresh does not happen too frequently. Valid suffixes include **s**, **m**, and **h** and are described in the Go **time** package. The default value is:
		
		`kubeProxyConfig:`
		`  proxyArguments:`
		`    iptables-min-sync-period:`
		`    - 0s`

4.16. DEPLOYING THE CLUSTER

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

When you have configured your VMC environment for OpenShift Container Platform deployment, you use the OpenShift Container Platform installation program from the bastion management host that is co-located in the VMC environment. The installation program and control plane automates the process of deploying and managing the resources needed for the OpenShift Container Platform cluster.

IMPORTANT

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

Prerequisites

- Configure an account with the cloud platform that hosts your cluster.
- Obtain the OpenShift Container Platform installation program and the pull secret for your cluster.
- Verify 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.

Procedure

- Change to the directory that contains the installation program and initialize the cluster deployment:

```
$ ./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**.

IMPORTANT

Use the **openshift-install** command from the bastion hosted in the VMC environment.

NOTE

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

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.

4.17. INSTALLING THE OPENSHIFT CLI BY DOWNLOADING THE BINARY

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

IMPORTANT

If you installed an earlier version of **oc**, you cannot use it to complete all of the commands in OpenShift Container Platform 4.12. Download and install the new version of **oc**.

Installing the OpenShift CLI on Linux

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

Procedure

1. Navigate to the OpenShift Container Platform downloads page on the Red Hat Customer Portal.

2. Select the architecture from the **Product Variant** drop-down list.

3. Select the appropriate version from the **Version** drop-down list.

4. Click **Download Now** next to the **OpenShift v4.12 Linux Client** 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>
```

Installing the OpenShift CLI on Windows

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

Procedure

1. Navigate to the OpenShift Container Platform downloads page on the Red Hat Customer Portal.
2. Select the appropriate version from the **Version** drop-down list.
3. Click **Download Now** next to the **OpenShift v4.12 Windows Client** entry and save the file.
4. Unzip the archive with a ZIP program.
5. Move the **oc** binary to a directory that is on your **PATH**.
   To check your **PATH**, open the command prompt and execute the following command:

   ```
   C:\> path
   ```

Verification

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

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

Installing the OpenShift CLI on macOS

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

Procedure

1. Navigate to the OpenShift Container Platform downloads page on the Red Hat Customer Portal.
2. Select the appropriate version from the **Version** drop-down list.
3. Click **Download Now** next to the **OpenShift v4.12 macOS Client** entry and save the file.

   > **NOTE**
   >
   > For macOS arm64, choose the **OpenShift v4.12 macOS arm64 Client** entry.

4. Unpack and unzip the archive.
5. Move the **oc** binary to a directory on your PATH.
   To check your **PATH**, open a terminal and execute the following command:

   ```
   $ echo $PATH
   ```

Verification

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

```
$ oc <command>
```

4.18. LOGGING IN TO THE CLUSTER BY USING THE CLI

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

Prerequisites

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

Procedure

1. Export the **kubeadmin** credentials:

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

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

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

```
$ oc whoami
```

Example output

```
system:admin
```

4.19. CREATING REGISTRY STORAGE

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

4.19.1. 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.

4.19.2. 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.

Instructions are shown for configuring a persistent volume, which is required for production clusters. Where applicable, instructions are shown for configuring an empty directory as the storage location, which is available for only non-production clusters.

Additional instructions are provided for allowing the image registry to use block storage types by using the **Recreate** rollout strategy during upgrades.

4.19.2.1. 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. To configure your registry to use storage, change the **spec.storage.pvc** 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:

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

Example output

```
No resourses 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:

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

Example output

```
storage:
  pvc:
    claim: (1)
```

(1) 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:

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

Example output

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

4.19.2.2. 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 PV for the block storage device, and create a 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 1
  namespace: openshift-image-registry 2
spec:
  accessModes:
  - ReadWriteOnce 3
  resources:
    requests:
      storage: 100Gi 4
```

1. A unique name that represents the **PersistentVolumeClaim** object.
2. The namespace for the **PersistentVolumeClaim** object, which is **openshift-image-registry**.
3. The access mode of the persistent volume claim. With **ReadWriteOnce**, the volume can be mounted with read and write permissions by a single node.
4. 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: 1
```

1. 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.

4.20. BACKING UP VMWARE VSPHERE VOLUMES

OpenShift Container Platform provisions new volumes as independent persistent disks to freely attach and detach the volume on any node in the cluster. As a consequence, it is not possible to back up volumes that use snapshots, or to restore volumes from snapshots. See Snapshot Limitations for more information.

Procedure

To create a backup of persistent volumes:

1. Stop the application that is using the persistent volume.
2. Clone the persistent volume.
3. Restart the application.
4. Create a backup of the cloned volume.
5. Delete the cloned volume.

4.21. TELEMETRY ACCESS FOR OPENSHIFT CONTAINER PLATFORM

In OpenShift Container Platform 4.12, the Telemetry service, which runs by default to provide metrics about cluster health and the success of updates, requires internet access. If your cluster is connected to the internet, Telemetry runs automatically, and your cluster is registered to OpenShift Cluster Manager Hybrid Cloud Console.

After you confirm that your OpenShift Cluster Manager Hybrid Cloud Console 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.

Additional resources

- See About remote health monitoring for more information about the Telemetry service

4.22. SERVICES FOR AN EXTERNAL LOAD BALANCER

You can configure an OpenShift Container Platform cluster to use an external load balancer in place of the default load balancer.

IMPORTANT

Configuring an external 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 an external load balancer:

- Ingress Controller
- OpenShift API
- OpenShift MachineConfig API

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

Figure 4.1. Example network workflow that shows an Ingress Controller operating in an OpenShift Container Platform environment

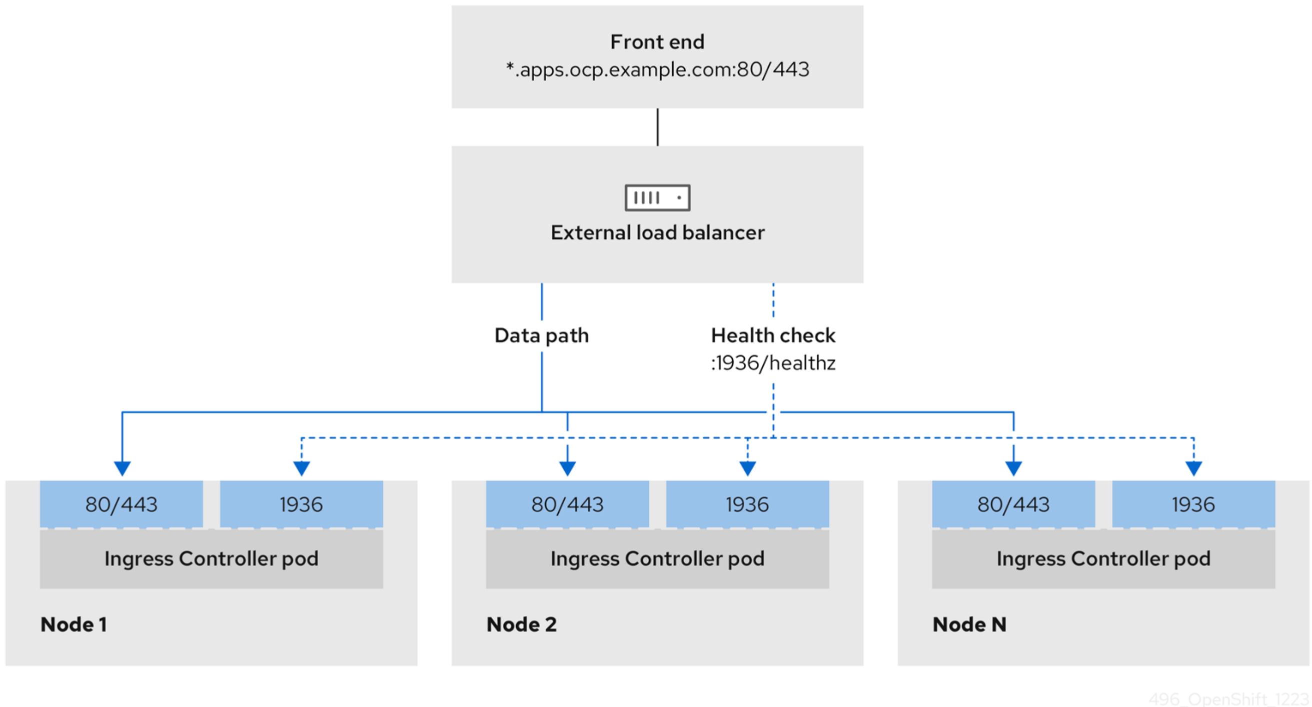


Figure 4.2. Example network workflow that shows an OpenShift API operating in an OpenShift Container Platform environment

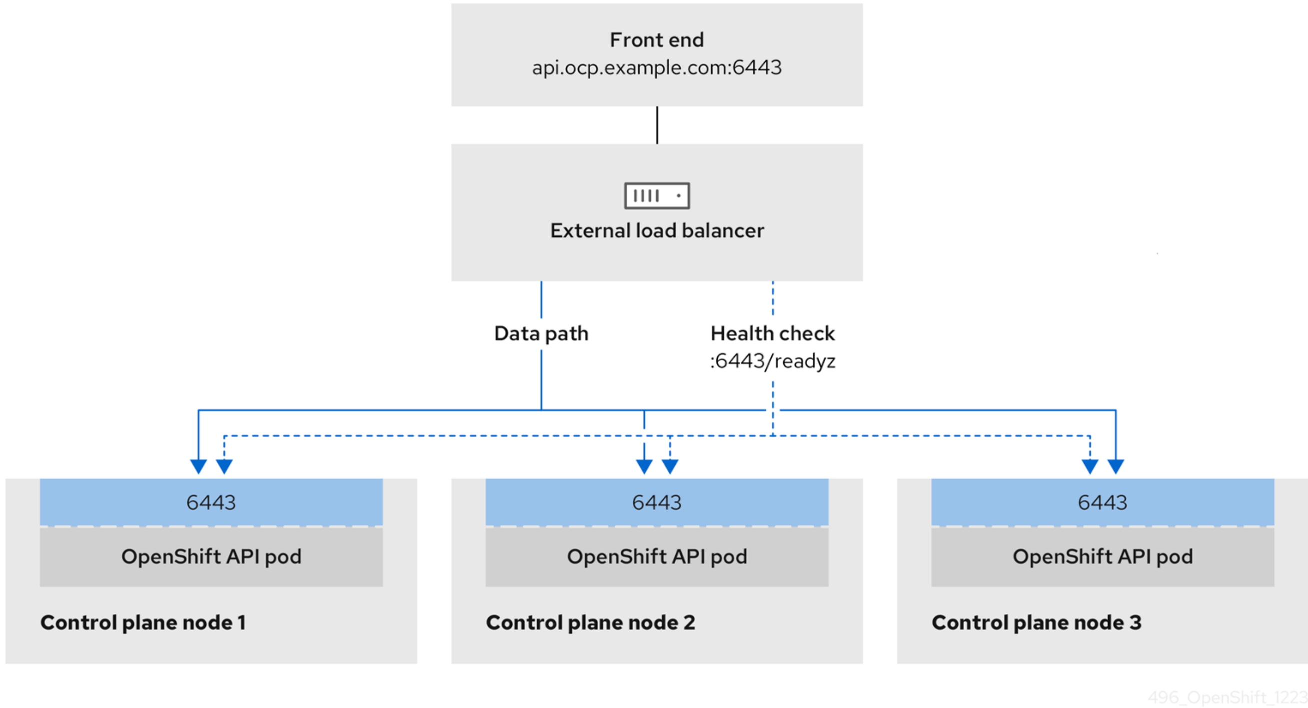
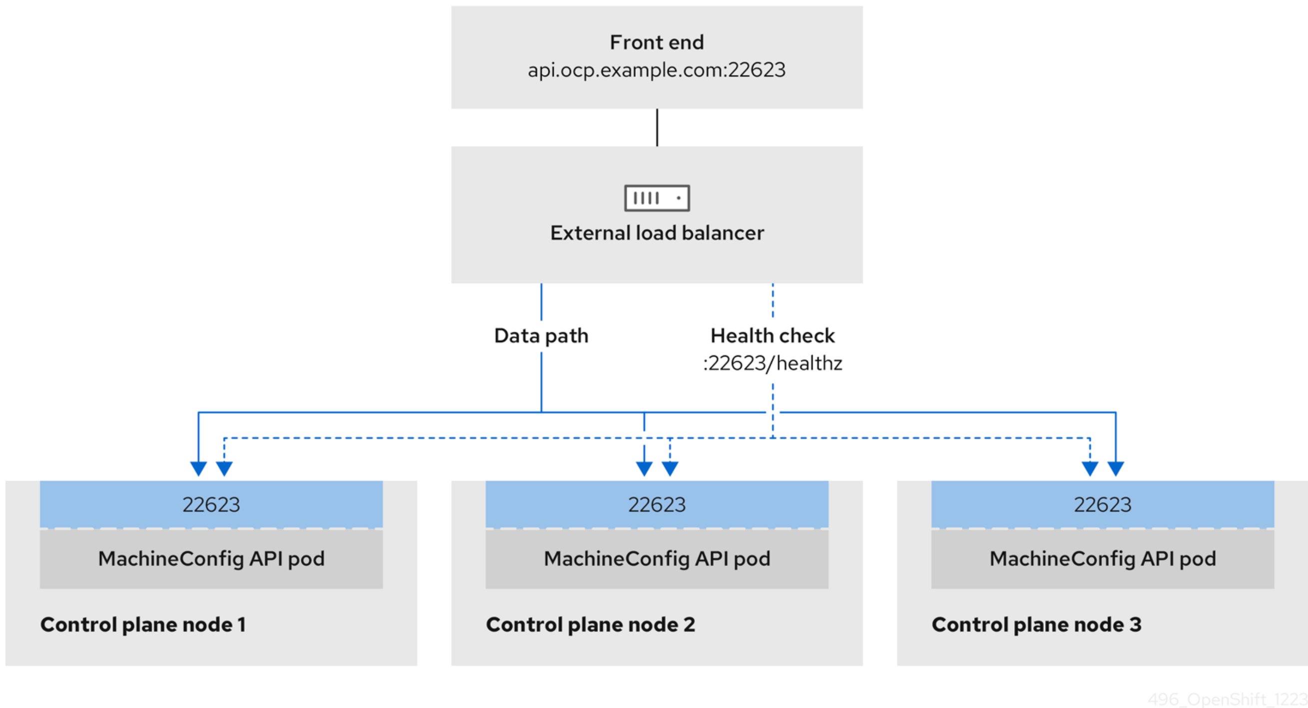


Figure 4.3. Example network workflow that shows an OpenShift MachineConfig API operating in an OpenShift Container Platform environment



The following configuration options are supported for external 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 an external 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 external 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 external 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.

4.22.1. Configuring an external load balancer

You can configure an OpenShift Container Platform cluster to use an external load balancer in place of the default load balancer.

IMPORTANT

Before you configure an external load balancer, ensure that you read the "Services for an external load balancer" section.

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

NOTE

MetalLB, that runs on a cluster, functions as an external load balancer.

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.

Ingress Controller prerequisites

- You defined a front-end IP address.
- TCP ports 443 and 80 are exposed on the front-end IP address of your load balancer.
- The front-end IP address, port 80 and port 443 are be 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 to 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.

Prerequisite 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 examples demonstrate health check specifications for the previously listed backend services:

Example of a Kubernetes API health check specification

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

Example of a Machine Config API health check specification

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

Example of an Ingress Controller health check specification

```
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, 443, and 80:

Example HAProxy configuration

```
#...
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
# ...
```

2. Use the **curl** CLI command to verify that the external 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=UlYWOyQ62LWjw2h003xtYSKlh1a0Py2hhctw0WmV2YEdhJjFyQwWcGBsja261dG
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 external load balancer. You must update records to your DNS server for the cluster API and applications over the load balancer.

Examples of 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. Use the **curl** CLI command to verify that the external 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, 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/JuLJMeoKNXlfiVv2YgZc09c3TBOBU4NI6kDXaJH1LdicNhN1UsQ
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=UlYWOyQ62LWjw2h003xtYSKlh1a0Py2hhctw0WmV2YEdhJjFyQwWcGBsja261dG
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
```

4.23. NEXT STEPS

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

CHAPTER 5. INSTALLING A CLUSTER ON VMC IN A RESTRICTED NETWORK

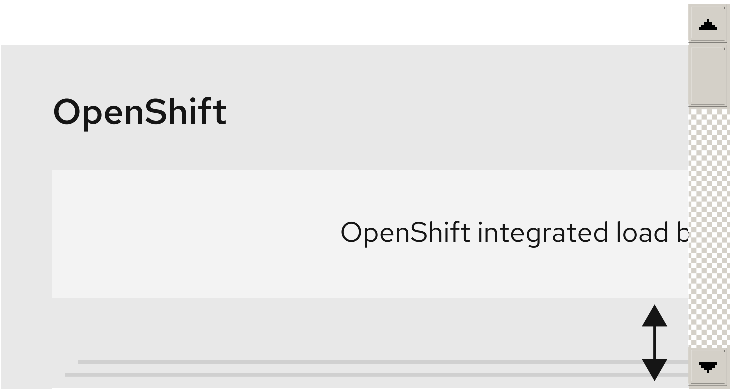
In OpenShift Container Platform version 4.12, you can install a cluster on VMware vSphere infrastructure in a restricted network by deploying it to VMware Cloud (VMC) on AWS.

Once you configure your VMC environment for OpenShift Container Platform deployment, you use the OpenShift Container Platform installation program from the bastion management host, co-located in the VMC environment. The installation program and control plane automates the process of deploying and managing the resources needed for the OpenShift Container Platform cluster.

> **NOTE**
>
> OpenShift Container Platform supports deploying a cluster to a single VMware vCenter only. Deploying a cluster with machines/machine sets on multiple vCenters is not supported.

5.1. SETTING UP VMC FOR VSPHERE

You can install OpenShift Container Platform on VMware Cloud (VMC) on AWS hosted vSphere clusters to enable applications to be deployed and managed both on-premise and off-premise, across the hybrid cloud.



You must configure several options in your VMC environment prior to installing OpenShift Container Platform on VMware vSphere. Ensure your VMC environment has the following prerequisites:

- Create a non-exclusive, DHCP-enabled, NSX-T network segment and subnet. Other virtual machines (VMs) can be hosted on the subnet, but at least eight IP addresses must be available for the OpenShift Container Platform deployment.
- Allocate two IP addresses, outside the DHCP range, and configure them with reverse DNS records.
  - A DNS record for **api.<cluster_name>.<base_domain>** pointing to the allocated IP address.
  - A DNS record for ***.apps.<cluster_name>.<base_domain>** pointing to the allocated IP address.
- Configure the following firewall rules:
  - An ANY:ANY firewall rule between the installation host and the software-defined data center (SDDC) management network on port 443. This allows you to upload the Red Hat Enterprise Linux CoreOS (RHCOS) OVA during deployment.

Enterprise Linux CoreOS (RHCOS) OVA during deployment.

- An HTTPS firewall rule between the OpenShift Container Platform compute network and vCenter. This connection allows OpenShift Container Platform to communicate with vCenter for provisioning and managing nodes, persistent volume claims (PVCs), and other resources.

- You must have the following information to deploy OpenShift Container Platform:
  - The OpenShift Container Platform cluster name, such as **vmc-prod-1**.
  - The base DNS name, such as **companyname.com**.
  - If not using the default, the pod network CIDR and services network CIDR must be identified, which are set by default to **10.128.0.0/14** and **172.30.0.0/16**, respectively. These CIDRs are used for pod-to-pod and pod-to-service communication and are not accessible externally; however, they must not overlap with existing subnets in your organization.
  - The following vCenter information:
    - vCenter hostname, username, and password
    - Datacenter name, such as **SDDC-Datacenter**
    - Cluster name, such as **Cluster-1**
    - Network name
    - Datastore name, such as **WorkloadDatastore**

    NOTE

    It is recommended to move your vSphere cluster to the VMC **Compute-ResourcePool** resource pool after your cluster installation is finished.

- A Linux-based host deployed to VMC as a bastion.
  - The bastion host can be Red Hat Enterprise Linux (RHEL) or any another Linux-based host; it must have internet connectivity and the ability to upload an OVA to the ESXi hosts.
  - Download and install the OpenShift CLI tools to the bastion host.
    - The **openshift-install** installation program
    - The OpenShift CLI (**oc**) tool

NOTE

You cannot use the VMware NSX Container Plugin for Kubernetes (NCP), and NSX is not used as the OpenShift SDN. The version of NSX currently available with VMC is incompatible with the version of NCP certified with OpenShift Container Platform.

However, the NSX DHCP service is used for virtual machine IP management with the full-stack automated OpenShift Container Platform deployment and with nodes provisioned, either manually or automatically, by the Machine API integration with vSphere. Additionally, NSX firewall rules are created to enable access with the OpenShift Container Platform cluster and between the bastion host and the VMC vSphere hosts.

5.1.1. VMC Sizer tool

VMware Cloud on AWS is built on top of AWS bare metal infrastructure; this is the same bare metal infrastructure which runs AWS native services. When a VMware cloud on AWS software-defined data center (SDDC) is deployed, you consume these physical server nodes and run the VMware ESXi hypervisor in a single tenant fashion. This means the physical infrastructure is not accessible to anyone else using VMC. It is important to consider how many physical hosts you will need to host your virtual infrastructure.

To determine this, VMware provides the VMC on AWS Sizer. With this tool, you can define the resources you intend to host on VMC:

- Types of workloads
- Total number of virtual machines
- Specification information such as:
  - Storage requirements
  - vCPUs
  - vRAM
  - Overcommit ratios

With these details, the sizer tool can generate a report, based on VMware best practices, and recommend your cluster configuration and the number of hosts you will need.

5.2. VSPHERE PREREQUISITES

- 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 block registry storage. For more information on persistent storage, see Understanding persistent storage.
- 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.

5.3. ABOUT INSTALLATIONS IN RESTRICTED NETWORKS

In OpenShift Container Platform 4.12, 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.

5.3.1. 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.

5.4. INTERNET ACCESS FOR OPENSHIFT CONTAINER PLATFORM

In OpenShift Container Platform 4.12, you require access to the internet to obtain the images that are necessary to install your cluster.

You must have internet access to:

- Access OpenShift Cluster Manager Hybrid Cloud Console 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.

5.5. 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 7.0 Update 2 or later
- Version 8.0 Update 1 or later

You can host the VMware vSphere infrastructure on-premise or on a VMware Cloud Verified provider that meets the requirements outlined in the following table:

Table 5.1. Version requirements for vSphere virtual environments

Virtual environment product	Required version
VMware virtual hardware	15 or later
vSphere ESXi hosts	7.0 Update 2 or later; 8.0 Update 1 or later
vCenter host	7.0 Update 2 or later; 8.0 Update 1 or later

IMPORTANT

Installing a cluster on VMware vSphere versions 7.0 and 7.0 Update 1 is deprecated. These versions are still fully supported, but all vSphere 6.x versions are no longer supported. Version 4.12 of OpenShift Container Platform requires VMware virtual hardware version 15 or later. To update the hardware version for your vSphere virtual machines, see the "Updating hardware on nodes running in vSphere" article in the *Updating clusters* section.

Table 5.2. Minimum supported vSphere version for VMware components

Component	Minimum supported versions	Description
Hypervisor	vSphere 7.0 Update 2 (or later) or vSphere 8.0 Update 1 (or later) with virtual hardware version 15	This 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.
Storage with in-tree drivers	vSphere 7.0 Update 2 or later; 8.0 Update 1 or later	This plugin creates vSphere storage by using the in-tree storage drivers for vSphere included in OpenShift Container Platform.

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.

5.6. 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 5.3. 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
	10256	openshift-sdn
UDP	**4789**	virtual extensible LAN (VXLAN)
	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	IPsec Encapsulating Security Payload (ESP)

Table 5.4. Ports used for all-machine to control plane communications

Protocol	Port	Description
TCP	**6443**	Kubernetes API

Table 5.5. Ports used for control plane machine to control plane machine communications

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

5.7. VMWARE VSPHERE CSI DRIVER OPERATOR REQUIREMENTS

To install the vSphere CSI Driver Operator, the following requirements must be met:

- VMware vSphere version: 7.0 Update 2 or later; 8.0 Update 1 or later
- vCenter version: 7.0 Update 2 or later; 8.0 Update 1 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.

Additional resources

- To remove a third-party 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.

5.8. VCENTER REQUIREMENTS

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

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. While most of the privileges are always required, some 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 amend vSphere roles for the specified objects to grant the required privileges.

An additional role is required if the installation program is to create a vSphere virtual machine folder.

Example 5.1. 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 will be created in the cluster root	**Host.Config.Storage** **Resource.AssignVMToPool** **VApp.AssignResourcePool** **VApp.Import** **VirtualMachine.Config.AddNewDisk**
vSphere vCenter Resource Pool	If an existing resource pool is provided	**Host.Config.Storage** **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.Add**

vSphere object for role	When required	Required privileges in vSphere API
		RemoveDevice 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 VirtualMachine.Config.ResetGuestInfo VirtualMachine.Config.Resource VirtualMachine.Config.Settings 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 Datacenter	If the installation program creates the virtual machine folder. For UPI, **VirtualMachine.Inventory.Create** and **VirtualMachine.Inventory.Delete** privileges are optional if your cluster does not use the Machine API.	InventoryService.Tagging.ObjectAttachable Resource.AssignVMToPool VApp.Import VirtualMachine.Config.AddExistingDisk VirtualMachine.Config.AddNewDisk VirtualMachine.Config.Add

vSphere object for role	When required	Required privileges in vSphere API
		RemoveDevice 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 VirtualMachine.Config.ResetGuestInfo VirtualMachine.Config.Resource VirtualMachine.Config.Settings 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.DeployTemplate VirtualMachine.Provisioning.MarkAsTemplate Folder.Create Folder.Delete

Example 5.2. Roles and privileges required for installation in vCenter graphical user interface (GUI)

vSphere object for role	When required	Required privileges in vCenter GUI
vSphere 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" "Profile-driven storage"."Profile-driven storage update" "Profile-driven storage"."Profile-driven storage view"**
vSphere vCenter Cluster	If VMs will be created 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 an existing resource pool is provided	**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 object for role	When required	Required privileges in vCenter GUI
vSphere Datastore	Always	**Datastore."Allocate space"** **Datastore."Browse datastore"** **Datastore."Low level file operations"** **"vSphere Tagging"."Assign or Unassign vSphere Tag on Object"**
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"**

vSphere object for role	When required	Required privileges in vCenter GUI
		"Virtual machine"."Change Configuration"."Rename" **"Virtual machine"."Change Configuration"."Reset guest information"** **"Virtual machine"."Change Configuration"."Change resource"** **"Virtual machine"."Change Configuration"."Change Settings"** **"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 Datacenter	If the installation program creates the virtual machine folder. For UPI, **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**

vSphere object for role	When required	Required privileges in vCenter GUI
		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 Settings" "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

vSphere object for role	When required	Required privileges in vCenter GUI
		Inventory"."Create new" "Virtual machine"."Edit Inventory"."Create from existing" "Virtual machine"."Edit Inventory"."Remove" "Virtual machine".Provisioning."Clone virtual machine" "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 propogate the permissions to child objects. These settings vary depending on whether or not you install the cluster into an existing folder.

Example 5.3. Required permissions and propagation settings

vSphere object	When required	Propagate to children	Permissions required
vSphere vCenter	Always	False	Listed required privileges
vSphere vCenter Datacenter	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 object	When required	Propagate to children	Permissions required
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.

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.

- OpenShift Container Platform generally supports compute-only vMotion, 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.

- Using Storage vMotion can cause issues and is not supported. If you are using 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 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.

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 is destroyed during the cluster installation process. A minimum of 800 GB of storage is required to use a standard cluster.

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

Cluster limits

Available resources vary between clusters. The number of possible clusters within a vCenter is limited 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.

Networking requirements

You must use the Dynamic Host Configuration Protocol (DHCP) for the network and ensure that the DHCP server is configured to provide persistent IP addresses to the cluster machines. In the DHCP lease, you must configure the DHCP to use the default gateway. All nodes must be in the same VLAN. You cannot scale the cluster using a second VLAN as a Day 2 operation. 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. Additionally, you must create the following networking resources before you install the OpenShift Container Platform cluster:

> **NOTE**
>
> It is recommended that each OpenShift Container Platform node in the cluster must have access to a Network Time Protocol (NTP) server that is discoverable via DHCP. Installation is possible without an NTP server. However, asynchronous server clocks will cause errors, which NTP server prevents.

Required IP Addresses

An installer-provisioned vSphere installation requires two static IP addresses:

- The **API** address is used to access the cluster API.
- The **Ingress** address is used for cluster ingress traffic.

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

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 5.6. Required DNS records

Component	Record	Description
API VIP	**api.<cluster_name>.<base_domain>.**	This DNS A/AAAA or CNAME 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.

Component	Record	Description
Ingress VIP	***.apps.<cluster_name>.<base_domain>.**	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.

5.9. GENERATING A KEY PAIR FOR CLUSTER NODE SSH ACCESS

During an OpenShift Container Platform installation, you can provide an SSH public key to the installation program. The key is passed to the Red Hat Enterprise Linux CoreOS (RHCOS) nodes through their Ignition config files and is used to authenticate SSH access to the nodes. The key is added to the **~/.ssh/authorized_keys** list for the **core** user on each node, which enables password-less authentication.

After the key is passed to the nodes, 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 such as AWS key pairs.

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> 1
```

1 Specify 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 FIPS validated or Modules In Process cryptographic libraries on 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> (1)
```

(1) Specify 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.

5.10. 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

1. 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.

2. 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
```

5.11. 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 4.12 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.

5.12. VMWARE VSPHERE REGION AND ZONE ENABLEMENT

You can deploy an OpenShift Container Platform cluster to multiple vSphere datacenters that run in a single VMware vCenter. Each datacenter 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**
>
> VMware vSphere region and zone enablement 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.

The default installation configuration deploys a cluster to a single vSphere datacenter. If you want to deploy a cluster to multiple vSphere datacenters, 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 datacenters 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 datacenter.

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

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

Table 5.7. Example of a configuration with multiple vSphere datacenters that run in a single VMware vCenter

Datacenter (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

5.13. CREATING THE INSTALLATION CONFIGURATION FILE

You can customize the OpenShift Container Platform cluster you install on VMware vSphere.

Prerequisites

- Obtain 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.
- Have the **imageContentSources** values that were generated during mirror registry creation.
- Obtain the contents of the certificate for your mirror registry.
- Retrieve a Red Hat Enterprise Linux CoreOS (RHCOS) image and upload it to an accessible location.
- Obtain service principal permissions at the subscription level.

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> 1
   ```

   1 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.

vi. Select the default vCenter datastore to use.

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.

xiii. Paste the pull secret from the Red Hat OpenShift Cluster Manager .

2. 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
```

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-----
  ZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZ
  -----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.

4. Make any other modifications to the **install-config.yaml** file that you require. You can find more information about the available parameters in the **Installation configuration parameters** section.

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.

5.13.1. Installation configuration parameters

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

NOTE

After installation, you cannot modify these parameters in the **install-config.yaml** file.

5.13.1.1. Required configuration parameters

Required installation configuration parameters are described in the following table:

Table 5.8. Required parameters

Parameter	Description	Values
apiVersion	The API version for the **install-config.yaml** content. The current version is **v1**. The installation program may also support older API versions.	String

Parameter	Description	Values
baseDomain	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.	A fully-qualified domain or subdomain name, such as **example.com**.
metadata	Kubernetes resource **ObjectMeta**, from which only the **name** parameter is consumed.	Object
metadata.name	The name of the cluster. DNS records for the cluster are all subdomains of **{{.metadata.name}}.{{.baseDomain}}**.	String of lowercase letters and hyphens (**-**), such as **dev**.
platform	The configuration for the specific platform upon which to perform the installation: **alibabacloud**, **aws**, **baremetal**, **azure**, **gcp**, **ibmcloud**, **nutanix**, **openstack**, **ovirt**, **vsphere**, or **{}**. For additional information about **platform.<platform>** parameters, consult the table for your specific platform that follows.	Object

Parameter	Description	Values
pullSecret	Get a pull secret from the Red Hat OpenShift Cluster Manager to authenticate downloading container images for OpenShift Container Platform components from services such as Quay.io.	`{ "auths":{ "cloud.openshift.com":{ "auth":"b3Blb=", "email":"you@example.com" }, "quay.io":{ "auth":"b3Blb=", "email":"you@example.com" } } }`

5.13.1.2. 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 provide different IP address blocks than the defaults.

Only IPv4 addresses are supported.

> **NOTE**
>
> Globalnet is not supported with Red Hat OpenShift Data Foundation disaster recovery solutions. For regional disaster recovery scenarios, ensure that you use a nonoverlapping range of private IP addresses for the cluster and service networks in each cluster.

Table 5.9. Network parameters

Parameter	Description	Values
networking	The configuration for the cluster network.	Object **NOTE** You cannot modify parameters specified by the **networking** object after installation.

Parameter	Description	Values
networking.networkType	The Red Hat OpenShift Networking network plugin to install.	Either **OpenShiftSDN** or **OVNKubernetes**. **OpenShiftSDN** is a CNI plugin for all-Linux networks. **OVNKubernetes** is a CNI plugin for Linux networks and hybrid networks that contain both Linux and Windows servers. The default value is **OVNKubernetes**.

networking.clusterNetwork	The IP address blocks for pods.
	
	The default value is **10.128.0.0/14** with a host prefix of **/23**.
	
	If you specify multiple IP address blocks, the blocks must not overlap.
		An array of objects. For example:
		
		`networking:`
		`  clusterNetwork:`
		`  - cidr: 10.128.0.0/14`
		`    hostPrefix: 23`

networking.clusterNetwork.cidr	Required if you use **networking.clusterNetwork**. An IP address block. An IPv4 network.	An IP address block in Classless Inter-Domain Routing (CIDR) notation. The prefix length for an IPv4 block is between **0** and **32**.
networking.clusterNetwork.hostPrefix	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.	A subnet prefix. The default value is **23**.

networking.serviceNetwork	The IP address block for services. The default value is **172.30.0.0/16**.
	
	The OpenShift SDN and OVN-Kubernetes network plugins support only a single IP address block for the service network.
		An array with an IP address block in CIDR format. For example:
		
		`networking:`
		`  serviceNetwork:`
		`   - 172.30.0.0/16`

networking.machineNetwork	The IP address blocks for machines.
	
	If you specify multiple IP address blocks, the blocks must not overlap.
		An array of objects. For example:
		
		`networking:`
		`  machineNetwork:`
		`  - cidr: 10.0.0.0/16`

Parameter	Description	Values
networking.machineNetwork.cidr	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. For libvirt, the default value is **192.168.126.0/24**.	An IP network block in CIDR notation. For example, **10.0.0.0/16**. NOTE Set the **networking.machineNetwork** to match the CIDR that the preferred NIC resides in.

5.13.1.3. Optional configuration parameters

Optional installation configuration parameters are described in the following table:

Table 5.10. Optional parameters

Parameter	Description	Values
additionalTrustBundle	A PEM-encoded X.509 certificate bundle that is added to the nodes' trusted certificate store. This trust bundle may also be used when a proxy has been configured.	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 *Installing*.	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**.	String
capabilities.additionalEnabledCapabilities	Extends the set of optional capabilities beyond what you specify in **baselineCapabilitySet**. You may specify multiple capabilities in this parameter.	String array
compute	The configuration for the machines that comprise the compute nodes.	Array of **MachinePool** objects.

Parameter	Description	Values
compute.architecture	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).	String
compute.hyperthreading	Whether to enable or disable simultaneous multithreading, or **hyperthreading**, on compute machines. By default, simultaneous multithreading is enabled to increase the performance of your machines' cores. IMPORTANT If you disable simultaneous multithreading, ensure that your capacity planning accounts for the dramatically decreased machine performance.	**Enabled** or **Disabled**
compute.name	Required if you use **compute**. The name of the machine pool.	**worker**
compute.platform	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.	**alibabacloud**, **aws**, **azure**, **gcp**, **ibmcloud**, **nutanix**, **openstack**, **ovirt**, **vsphere**, or **{}**
compute.replicas	The number of compute machines, which are also known as worker machines, to provision.	A positive integer greater than or equal to **2**. The default value is **3**.
featureSet	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".	String. The name of the feature set to enable, such as **TechPreviewNoUpgrade**.

Parameter	Description	Values
controlPlane	The configuration for the machines that comprise the control plane.	Array of **MachinePool** objects.
controlPlane.architecture	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).	String
controlPlane.hyperthreading	Whether to enable or disable simultaneous multithreading, or **hyperthreading**, on control plane machines. By default, simultaneous multithreading is enabled to increase the performance of your machines' cores. IMPORTANT If you disable simultaneous multithreading, ensure that your capacity planning accounts for the dramatically decreased machine performance.	**Enabled** or **Disabled**
controlPlane.name	Required if you use **controlPlane**. The name of the machine pool.	**master**
controlPlane.platform	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.	**alibabacloud**, **aws**, **azure**, **gcp**, **ibmcloud**, **nutanix**, **openstack**, **ovirt**, **vsphere**, or **{}**
controlPlane.replicas	The number of control plane machines to provision.	The only supported value is **3**, which is the default value.

Parameter	Description	Values
credentialsMode	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. NOTE Not all CCO modes are supported for all cloud providers. For more information about CCO modes, see the *Cloud Credential Operator* entry in the *Cluster Operators reference* content. NOTE If your AWS account has service control policies (SCP) enabled, you must configure the **credentialsMode** parameter to **Mint**, **Passthrough** or **Manual**.	**Mint**, **Passthrough**, **Manual** or an empty string (**""**).

Parameter	Description	Values
fips	Enable or disable FIPS mode. The default is **false** (disabled). 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 Installing the system in FIPS mode. The use of FIPS validated or Modules In Process cryptographic libraries is only supported on OpenShift Container Platform deployments on the **x86_64**, **ppc64le**, and **s390x** architectures. NOTE If you are using Azure File storage, you cannot enable FIPS mode.	**false** or **true**
imageContentSources	Sources and repositories for the release-image content.	Array of objects. Includes a **source** and, optionally, **mirrors**, as described in the following rows of this table.
imageContentSources.source	Required if you use **imageContentSources**. Specify the repository that users refer to, for example, in image pull specifications.	String

Parameter	Description	Values
imageContentSources.mirrors	Specify one or more repositories that may also contain the same images.	Array of strings
publish	How to publish or expose the user-facing endpoints of your cluster, such as the Kubernetes API, OpenShift routes.	**Internal** or **External**. The default value is **External**. Setting this field to **Internal** is not supported on non-cloud platforms. IMPORTANT If the value of the field is set to **Internal**, the cluster will become non-functional. For more information, refer to BZ#1953035.
sshKey	The SSH key to authenticate access to 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.	For example, **sshKey: ssh-ed25519 AAAA..**.

5.13.1.4. Additional VMware vSphere configuration parameters

Additional VMware vSphere configuration parameters are described in the following table.

> NOTE
>
> The **platform.vsphere** parameter prefixes each parameter listed in the table.

Table 5.11. Additional VMware vSphere cluster parameters

Parameter	Description	Values
vCenter	The fully-qualified hostname or IP address of the vCenter server.	String

Parameter	Description	Values
username	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.	String
password	The password for the vCenter user name.	String
datacenter	The name of the data center to use in the vCenter instance.	String
defaultDatastore	The name of the default datastore to use for provisioning volumes.	String
folder	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.	String, for example, **/<datacenter_name>/vm/<folder_name>/<subfolder_name>**.
resourcePool	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 **/<datacenter_name>/host/<cluster_name>/Resources**.	String, for example, **/<datacenter_name>/host/<cluster_name>/Resources/<resource_pool_name>/<optional_nested_resource_pool_name>**.
network	The network in the vCenter instance that contains the virtual IP addresses and DNS records that you configured.	String
cluster	The vCenter cluster to install the OpenShift Container Platform cluster in.	String
apiVIPs	The virtual IP (VIP) address that you configured for control plane API access. NOTE 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.	An IP address, for example **128.0.0.1**.

Parameter	Description	Values
ingressVIPs	The virtual IP (VIP) address that you configured for cluster ingress. NOTE 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.	An IP address, for example **128.0.0.1**.
diskType	Optional. The disk provisioning method. This value defaults to the vSphere default storage policy if not set.	Valid values are **thin**, **thick**, or **eagerZeroedThick**.

5.13.1.5. Optional VMware vSphere machine pool configuration parameters

Optional VMware vSphere machine pool configuration parameters are described in the following table.

NOTE

The **platform.vsphere** parameter prefixes each parameter listed in the table.

Table 5.12. Optional VMware vSphere machine pool parameters

Parameter	Description	Values
clusterOSImage	The location from which the installation program downloads the RHCOS image. You must set this parameter to perform an installation in a restricted network.	An HTTP or HTTPS URL, optionally with a SHA-256 checksum. For example, **https://mirror.openshift.com/images/rhcos-<version>-vmware.<architecture>.ova**.
osDisk.diskSizeGB	The size of the disk in gigabytes.	Integer
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.	Integer

Parameter	Description	Values
coresPerSocket	The number of cores per socket in a virtual machine. The number of virtual sockets on the virtual machine is **platform.vsphere.cpus**/**platform.vsphere.coresPerSocket**. The default value for control plane nodes and worker nodes is **4** and **2**, respectively.	Integer
memoryMB	The size of a virtual machine's memory in megabytes.	Integer

5.13.1.6. Region and zone enablement configuration parameters

To use the region and zone enablement feature, you must specify region and zone enablement parameters in your installation file.

IMPORTANT

Before you modify the **install-config.yaml** file to configure a region and zone enablement environment, read the "VMware vSphere region and zone enablement" and the "Configuring regions and zones for a VMware vCenter" sections.

NOTE

The **platform.vsphere** parameter prefixes each parameter listed in the table.

Table 5.13. Region and zone enablement parameters

Parameter	Description	Values
failureDomains	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.	String
failureDomains.name	The name of the failure domain. The machine pools use this name to reference the failure domain.	String
failureDomains.server	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.	String
failureDomains.region	You define a region by using a tag from the **openshift-region** tag category. The tag must be attached to the vCenter datacenter.	String

Parameter	Description	Values
failureDomains.zone	You define a zone by using a tag from the **openshift-zone** tag category. The tag must be attached to the vCenter datacenter.	String
failureDomains.topology.computeCluster	This parameter defines the compute cluster associated with the failure domain. If you do not define this parameter in your configuration, the compute cluster takes the value of **platform.vsphere.cluster** and **platform.vsphere.datacenter**.	String
failureDomains.topology.folder	The absolute path of an existing folder where the installation program creates the virtual machines. If you do not define this parameter in your configuration, the folder takes the value of **platform.vsphere.folder**.	String
failureDomains.topology.datacenter	Defines the datacenter where OpenShift Container Platform virtual machines (VMs) operate. If you do not define this parameter in your configuration, the datacenter defaults to **platform.vsphere.datacenter**.	String
failureDomains.topology.datastore	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.	String
failureDomains.topology.networks	Lists any network in the vCenter instance that contains the virtual IP addresses and DNS records that you configured. If you do not define this parameter in your configuration, the network takes the value of **platform.vsphere.network**.	String
failureDomains.topology.resourcePool	Optional: The absolute path of an existing resource pool where the installation program creates the virtual machines, for example, **/<datacenter_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 **/<datacenter_name>/host/<cluster_name>/Resources**.	String

5.13.2. 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 modify the values of the required parameters.

```
apiVersion: v1
baseDomain: example.com 1
compute: 2
  name: worker
  replicas: 3
  platform:
    vsphere: 3
      cpus: 2
      coresPerSocket: 2
      memoryMB: 8192
      osDisk:
        diskSizeGB: 120
controlPlane: 4
  name: master
  replicas: 3
  platform:
    vsphere: 5
      cpus: 4
      coresPerSocket: 2
      memoryMB: 16384
      osDisk:
        diskSizeGB: 120
metadata:
  name: cluster 6
platform:
  vsphere:
    vcenter: your.vcenter.server
    username: username
    password: password
    datacenter: datacenter
    defaultDatastore: datastore
    folder: folder
    resourcePool: resource_pool 7
    diskType: thin 8
    network: VM_Network
    cluster: vsphere_cluster_name 9
    apiVIPs:
      - api_vip
    ingressVIPs:
      - ingress_vip
    clusterOSImage: http://mirror.example.com/images/rhcos-47.83.202103221318-0-
vmware.x86_64.ova 10
fips: false
pullSecret: '{"auths":{"<local_registry>": {"auth": "<credentials>","email": "you@example.com"}}}' 11
sshKey: 'ssh-ed25519 AAAA...'
additionalTrustBundle: | 12
  -----BEGIN CERTIFICATE-----
  ZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZ
  -----END CERTIFICATE-----
imageContentSources: 13
- 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. Only one control plane pool is used.

3 5 Optional: Provide additional configuration for the machine pool parameters for the compute and control plane machines.

6 The cluster name that you specified in your DNS records.

7 Optional: Provide 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.

8 The vSphere disk provisioning method.

9 The vSphere cluster to install the OpenShift Container Platform cluster in.

10 The location of the Red Hat Enterprise Linux CoreOS (RHCOS) image that is accessible from the bastion server.

11 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.

12 Provide the contents of the certificate file that you used for your mirror registry.

13 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.

5.13.3. Configuring the cluster-wide proxy during installation

Production environments can deny direct access to the internet and instead have an HTTP or HTTPS proxy available. You can configure a new OpenShift Container Platform cluster to use a proxy by configuring the proxy settings in the **install-config.yaml** file.

Prerequisites

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

- You 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 Platform (GCP), 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> 1
  httpsProxy: https://<username>:<pswd>@<ip>:<port> 2
  noProxy: example.com 3
additionalTrustBundle: | 4
    -----BEGIN CERTIFICATE-----
    <MY_TRUSTED_CA_CERT>
    -----END CERTIFICATE-----
additionalTrustBundlePolicy: <policy_to_add_additionalTrustBundle> 5
```

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

2 A proxy URL to use for creating HTTPS connections outside the cluster.

3 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.

4 If provided, the installation program generates a config map that is named **user-ca-bundle** in the **openshift-config** namespace that contains one or more additional CA certificates that are required for proxying HTTPS connections. The Cluster Network Operator then creates a **trusted-ca-bundle** config map that merges these contents with the Red Hat Enterprise Linux CoreOS (RHCOS) trust bundle, and this config map is referenced in the **trustedCA** field of the **Proxy** object. The **additionalTrustBundle** field is required unless the proxy's identity certificate is signed by an authority from the RHCOS trust bundle.

5 Optional: The policy to determine 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**.

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.

5.13.4. Configuring regions and zones for a VMware vCenter

You can modify the default installation configuration file to deploy an OpenShift Container Platform cluster to multiple vSphere datacenters that run in a single VMware vCenter.

IMPORTANT

VMware vSphere region and zone enablement 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 .

IMPORTANT

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

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 datacenter objects for your VMware vCenter server. Consider specifying multiple failure domains if you need to provision virtual machine nodes in different datacenters, clusters, datastores, and other components. To enable regions and zones, you must define multiple failure domains for your OpenShift Container Platform cluster.

> **NOTE**
>
> You cannot change a failure domain after you installed an OpenShift Container Platform cluster on the VMware vSphere platform. You can add additional failure domains after cluster installation.

Procedure

1. Enter the following **govc** command-line tool commands to create the **openshift-region** and **openshift-zone** vCenter tag categories:

   > **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. To create a region tag for each region vSphere datacenter where you want to deploy your cluster, enter the following command in your terminal:

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

3. To create a zone tag for each vSphere cluster where you want to deploy your cluster, enter the following command:

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

4. Attach region tags to each vCenter datacenter object by entering the following command:

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

5. Attach the zone tags to each vCenter datacenter object by entering the following command:

   ```
   $ govc tags.attach -c <zone_tag_category> <zone_tag_1> /<datacenter_1>/host/vcs-mdcnc-workload-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 datacenters defined in a vSphere center

```
apiVersion: v1
baseDomain: example.com
featureSet: TechPreviewNoUpgrade 1
compute:
  name: worker
  replicas: 3
  vsphere:
    zones: 2
      - "<machine_pool_zone_1>"
      - "<machine_pool_zone_2>"
controlPlane:
  name: master
  replicas: 3
  vsphere:
    zones: 3
      - "<machine_pool_zone_1>"
      - "<machine_pool_zone_2>"
metadata:
  name: cluster
platform:
  vsphere:
    vcenter: <vcenter_server> 4
    username: <username> 5
    password: <password> 6
    datacenter: datacenter 7
    defaultDatastore: datastore 8
    folder: "/<datacenter_name>/vm/<folder_name>/<subfolder_name>" 9
    cluster: cluster 10
    resourcePool: "/<datacenter_name>/host/<cluster_name>/Resources/<resource_pool_name>" 11
    diskType: thin
    failureDomains: 12
    - name: <machine_pool_zone_1> 13
      region: <region_tag_1> 14
      zone: <zone_tag_1> 15
      topology: 16
        datacenter: <datacenter1> 17
        computeCluster: "/<datacenter1>/host/<cluster1>" 18
        resourcePool: "/<datacenter1>/host/<cluster1>/Resources/<resourcePool1>" 19
        networks: 20
        - <VM_Network1_name>
        datastore: "/<datacenter1>/datastore/<datastore1>" 21
    - name: <machine_pool_zone_2>
      region: <region_tag_2>
      zone: <zone_tag_2>
      topology:
        datacenter: <datacenter2>
        computeCluster: "/<datacenter2>/host/<cluster2>"
        networks:
        - <VM_Network2_name>
        datastore: "/<datacenter2>/datastore/<datastore2>"
```

```
        resourcePool: "/<datacenter2>/host/<cluster2>/Resources/<resourcePool2>"
        folder: "/<datacenter2>/vm/<folder2>"
# ...
```

1 You must define set the **TechPreviewNoUpgrade** as the value for this parameter, so that you can use the VMware vSphere region and zone enablement feature.

2 3 An optional parameter for specifying a vCenter cluster. You define a zone by using a tag from the **openshift-zone** tag category. If you do not define this parameter, nodes will be distributed among all defined failure-domains.

4 5 6 7 8 9 10 11 The default vCenter topology. The installation program uses this topology information to deploy the bootstrap node. Additionally, the topology defines the default datastore for vSphere persistent volumes.

12 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. If you do not define this parameter, the installation program uses the default vCenter topology.

13 Defines the name of the failure domain. Each failure domain is referenced in the **zones** parameter to scope a machine pool to the failure domain.

14 You define a region by using a tag from the **openshift-region** tag category. The tag must be attached to the vCenter datacenter.

15 You define a zone by using a tag from the **openshift-zone tag** category. The tag must be attached to the vCenter datacenter.

16 Specifies the vCenter resources associated with the failure domain.

17 An optional parameter for defining the vSphere datacenter that is associated with a failure domain. If you do not define this parameter, the installation program uses the default vCenter topology.

18 An optional parameter for stating the absolute file path for the compute cluster that is associated with the failure domain. If you do not define this parameter, the installation program uses the default vCenter topology.

19 An optional parameter for the installer-provisioned infrastructure. The parameter sets the absolute path of an existing resource pool where the installation program creates the virtual machines, for example, **/<datacenter_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_datacenter/host/example_cluster/Resources**.

20 An optional parameter that lists any network in the vCenter instance that contains the virtual IP addresses and DNS records that you configured. If you do not define this parameter, the installation program uses the default vCenter topology.

21 An optional parameter for specifying a datastore to use for provisioning volumes. If you do not define this parameter, the installation program uses the default vCenter topology.

5.14. DEPLOYING THE CLUSTER

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

When you have configured your VMC environment for OpenShift Container Platform deployment, you use the OpenShift Container Platform installation program from the bastion management host that is co-located in the VMC environment. The installation program and control plane automates the process of deploying and managing the resources needed for the OpenShift Container Platform cluster.

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

Prerequisites

- Configure an account with the cloud platform that hosts your cluster.
- Obtain the OpenShift Container Platform installation program and the pull secret for your cluster.
- Verify 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.

Procedure

- Change to the directory that contains the installation program and initialize the cluster deployment:

```
$ ./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**.

> **IMPORTANT**
>
> Use the **openshift-install** command from the bastion hosted in the VMC environment.

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

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.

5.15. INSTALLING THE OPENSHIFT CLI BY DOWNLOADING THE BINARY

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

> **IMPORTANT**
>
> If you installed an earlier version of **oc**, you cannot use it to complete all of the commands in OpenShift Container Platform 4.12. Download and install the new version of **oc**.

Installing the OpenShift CLI on Linux

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

Procedure

1. Navigate to the OpenShift Container Platform downloads page on the Red Hat Customer Portal.

2. Select the architecture from the **Product Variant** drop-down list.

3. Select the appropriate version from the **Version** drop-down list.

4. Click **Download Now** next to the **OpenShift v4.12 Linux Client** 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>
```

Installing the OpenShift CLI on Windows

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

Procedure

1. Navigate to the OpenShift Container Platform downloads page on the Red Hat Customer Portal.

2. Select the appropriate version from the **Version** drop-down list.

3. Click **Download Now** next to the **OpenShift v4.12 Windows Client** entry and save the file.

4. Unzip the archive with a ZIP program.

5. Move the **oc** binary to a directory that is on your **PATH**.
   To check your **PATH**, open the command prompt and execute the following command:

```
C:\> path
```

Verification

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

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

Installing the OpenShift CLI on macOS

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

Procedure

1. Navigate to the OpenShift Container Platform downloads page on the Red Hat Customer Portal.

2. Select the appropriate version from the **Version** drop-down list.

3. Click **Download Now** next to the **OpenShift v4.12 macOS Client** entry and save the file.

> **NOTE**
>
> For macOS arm64, choose the **OpenShift v4.12 macOS arm64 Client** entry.

4. Unpack and unzip the archive.

5. Move the **oc** binary to a directory on your PATH.
   To check your **PATH**, open a terminal and execute the following command:

```
$ echo $PATH
```

Verification

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

```
$ oc <command>
```

5.16. LOGGING IN TO THE CLUSTER BY USING THE CLI

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

Prerequisites

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

Procedure

1. Export the **kubeadmin** credentials:

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

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

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

```
$ oc whoami
```

Example output

```
system:admin
```

5.17. 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.

5.18. CREATING REGISTRY STORAGE

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

5.18.1. 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.

5.18.2. 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.

Instructions are shown for configuring a persistent volume, which is required for production clusters. Where applicable, instructions are shown for configuring an empty directory as the storage location, which is available for only non-production clusters.

Additional instructions are provided for allowing the image registry to use block storage types by using the **Recreate** rollout strategy during upgrades.

5.18.2.1. 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. To configure your registry to use storage, change the **spec.storage.pvc** 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:

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

   Example output

   ```
   No resourses 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:

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

   Example output

```
storage:
  pvc:
    claim: 1
```

1. 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:

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

Example output

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

5.19. TELEMETRY ACCESS FOR OPENSHIFT CONTAINER PLATFORM

In OpenShift Container Platform 4.12, the Telemetry service, which runs by default to provide metrics about cluster health and the success of updates, requires internet access. If your cluster is connected to the internet, Telemetry runs automatically, and your cluster is registered to OpenShift Cluster Manager Hybrid Cloud Console.

After you confirm that your OpenShift Cluster Manager Hybrid Cloud Console 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.

Additional resources

- See About remote health monitoring for more information about the Telemetry service

5.20. SERVICES FOR AN EXTERNAL LOAD BALANCER

You can configure an OpenShift Container Platform cluster to use an external load balancer in place of the default load balancer.

IMPORTANT

Configuring an external 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 an external load balancer:

- Ingress Controller

- OpenShift API
- OpenShift MachineConfig API

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

Figure 5.1. Example network workflow that shows an Ingress Controller operating in an OpenShift Container Platform environment

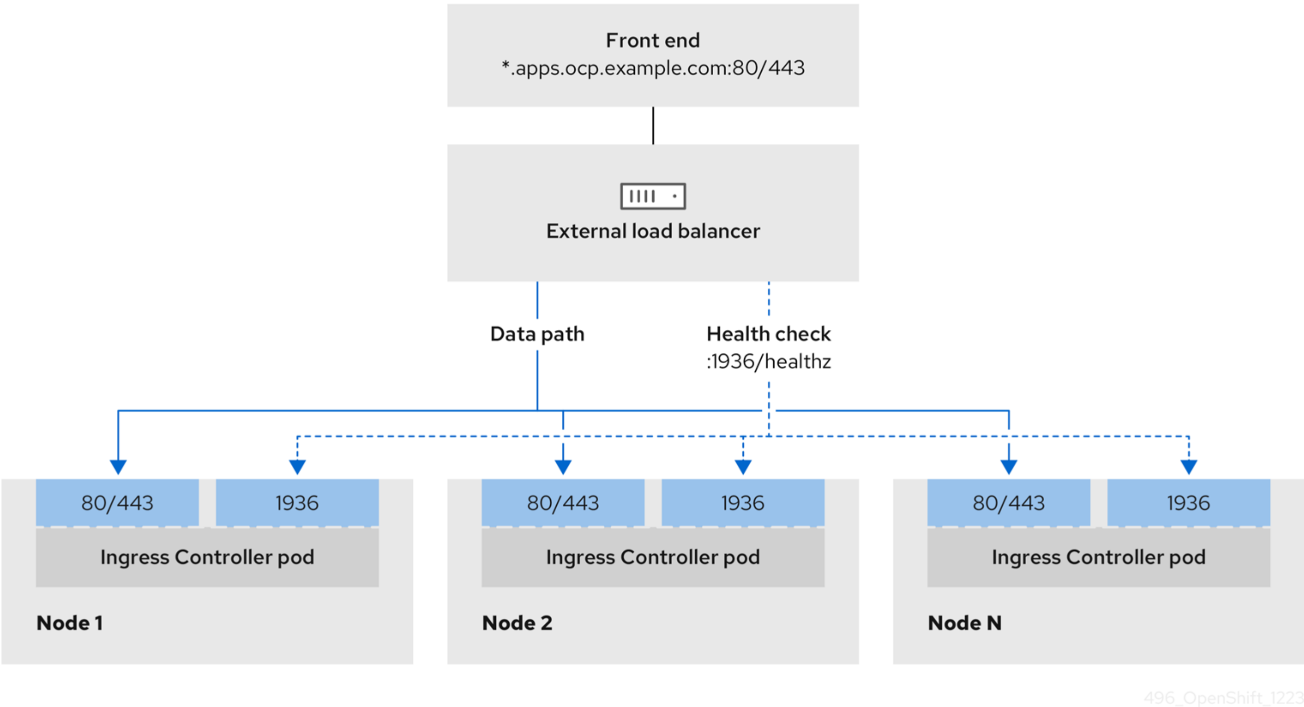


Figure 5.2. Example network workflow that shows an OpenShift API operating in an OpenShift Container Platform environment

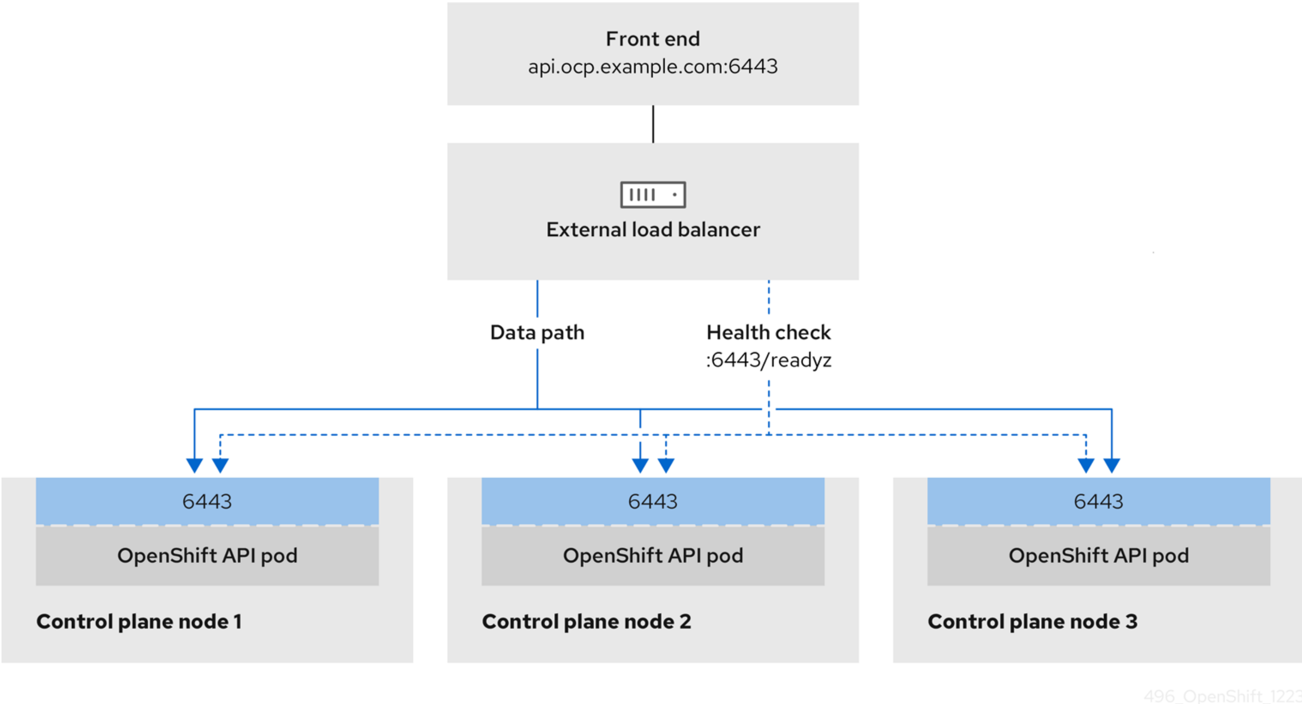
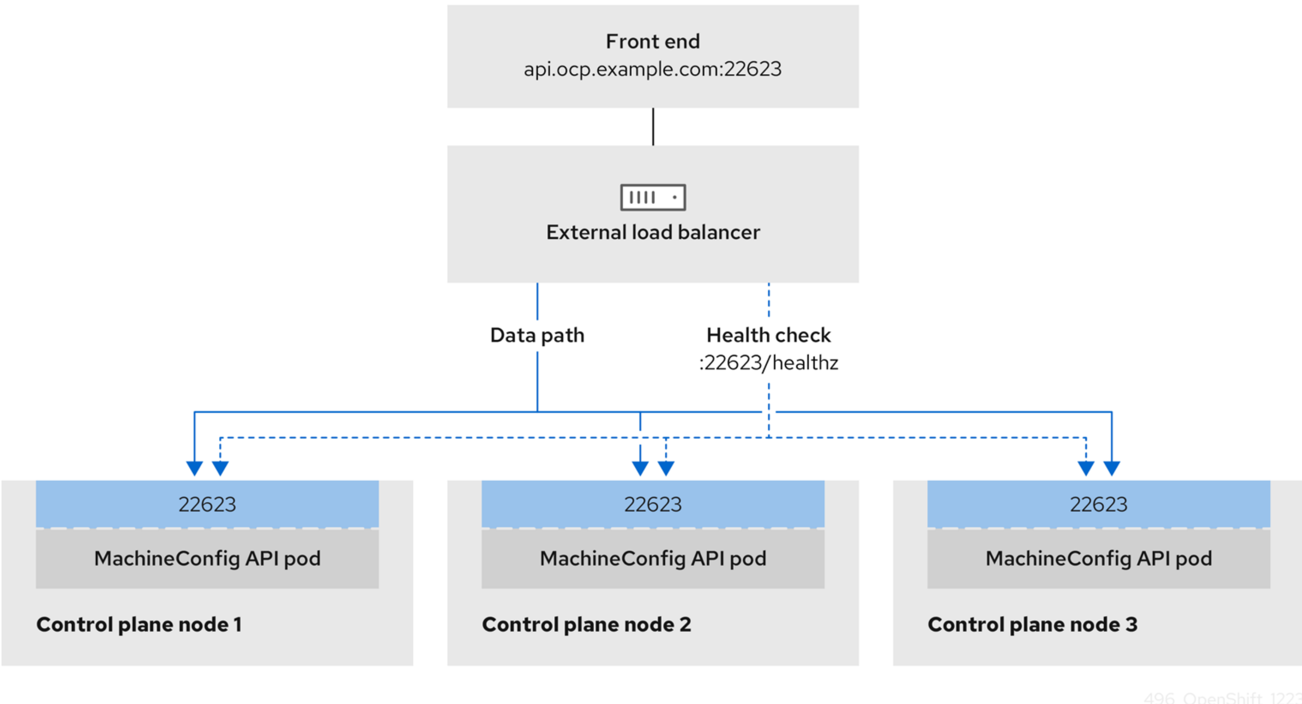


Figure 5.3. Example network workflow that shows an OpenShift MachineConfig API operating in an OpenShift Container Platform environment



The following configuration options are supported for external 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 an external 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 external 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 external 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.

5.20.1. Configuring an external load balancer

You can configure an OpenShift Container Platform cluster to use an external load balancer in place of the default load balancer.

IMPORTANT

Before you configure an external load balancer, ensure that you read the "Services for an external load balancer" section.

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

NOTE

MetalLB, that runs on a cluster, functions as an external load balancer.

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.

Ingress Controller prerequisites

- You defined a front-end IP address.
- TCP ports 443 and 80 are exposed on the front-end IP address of your load balancer.
- The front-end IP address, port 80 and port 443 are be 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 to 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.

Prerequisite 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 examples demonstrate health check specifications for the previously listed backend services:

Example of a Kubernetes API health check specification

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

Example of a Machine Config API health check specification

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

Example of an Ingress Controller health check specification

```
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, 443, and 80:

   Example HAProxy configuration

```
#...
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
# ...
```

2. Use the **curl** CLI command to verify that the external 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=UlYWOyQ62LWjw2h003xtYSKlh1a0Py2hhctw0WmV2YEdhJjFyQwWcGBsja261dG
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 external load balancer. You must update records to your DNS server for the cluster API and applications over the load balancer.

Examples of 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. Use the **curl** CLI command to verify that the external 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, 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/JuLJMeoKNXlfiVv2YgZc09c3TBOBU4NI6kDXaJH1LdicNhN1UsQ
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=UlYWOyQ62LWjw2h003xtYSKlh1a0Py2hhctw0WmV2YEdhJjFyQwWcGBsja261dG
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
```

5.21. NEXT STEPS

- Customize your cluster.
- Configure image streams for the Cluster Samples Operator and the **must-gather** tool.
- Learn how to use Operator Lifecycle Manager (OLM) on restricted networks .
- If necessary, you can opt out of remote health reporting .
- Set up your registry and configure registry storage .

CHAPTER 6. INSTALLING A CLUSTER ON VMC WITH USER-PROVISIONED INFRASTRUCTURE

In OpenShift Container Platform version 4.12, you can install a cluster on VMware vSphere infrastructure that you provision by deploying it to VMware Cloud (VMC) on AWS.

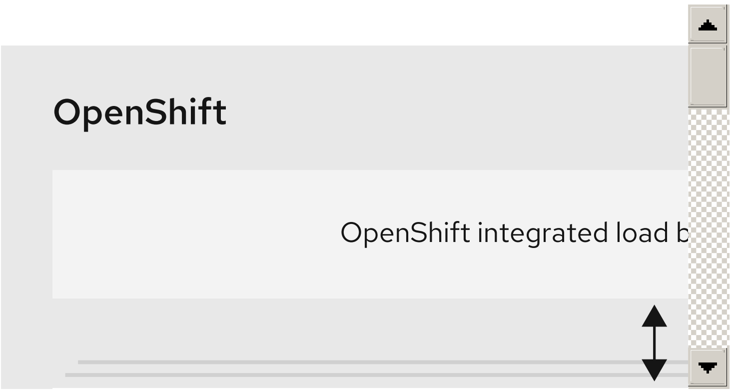
Once you configure your VMC environment for OpenShift Container Platform deployment, you use the OpenShift Container Platform installation program from the bastion management host, co-located in the VMC environment. The installation program and control plane automates the process of deploying and managing the resources needed for the OpenShift Container Platform cluster.

NOTE

OpenShift Container Platform supports deploying a cluster to a single VMware vCenter only. Deploying a cluster with machines/machine sets on multiple vCenters is not supported.

6.1. SETTING UP VMC FOR VSPHERE

You can install OpenShift Container Platform on VMware Cloud (VMC) on AWS hosted vSphere clusters to enable applications to be deployed and managed both on-premise and off-premise, across the hybrid cloud.



You must configure several options in your VMC environment prior to installing OpenShift Container Platform on VMware vSphere. Ensure your VMC environment has the following prerequisites:

- Create a non-exclusive, DHCP-enabled, NSX-T network segment and subnet. Other virtual machines (VMs) can be hosted on the subnet, but at least eight IP addresses must be available for the OpenShift Container Platform deployment.
- Configure the following firewall rules:
  - An ANY:ANY firewall rule between the OpenShift Container Platform compute network and the internet. This is used by nodes and applications to download container images.
  - An ANY:ANY firewall rule between the installation host and the software-defined data center (SDDC) management network on port 443. This allows you to upload the Red Hat Enterprise Linux CoreOS (RHCOS) OVA during deployment.
  - An HTTPS firewall rule between the OpenShift Container Platform compute network and vCenter. This connection allows OpenShift Container Platform to communicate with vCenter for provisioning and managing nodes, persistent volume claims (PVCs), and other resources.

- You must have the following information to deploy OpenShift Container Platform:
  - The OpenShift Container Platform cluster name, such as **vmc-prod-1**.
  - The base DNS name, such as **companyname.com**.
  - If not using the default, the pod network CIDR and services network CIDR must be identified, which are set by default to **10.128.0.0/14** and **172.30.0.0/16**, respectively. These CIDRs are used for pod-to-pod and pod-to-service communication and are not accessible externally; however, they must not overlap with existing subnets in your organization.
  - The following vCenter information:
    - vCenter hostname, username, and password
    - Datacenter name, such as **SDDC-Datacenter**
    - Cluster name, such as **Cluster-1**
    - Network name
    - Datastore name, such as **WorkloadDatastore**

    NOTE

    It is recommended to move your vSphere cluster to the VMC **Compute-ResourcePool** resource pool after your cluster installation is finished.
- A Linux-based host deployed to VMC as a bastion.
  - The bastion host can be Red Hat Enterprise Linux (RHEL) or any another Linux-based host; it must have internet connectivity and the ability to upload an OVA to the ESXi hosts.
  - Download and install the OpenShift CLI tools to the bastion host.
    - The **openshift-install** installation program
    - The OpenShift CLI (**oc**) tool

NOTE

You cannot use the VMware NSX Container Plugin for Kubernetes (NCP), and NSX is not used as the OpenShift SDN. The version of NSX currently available with VMC is incompatible with the version of NCP certified with OpenShift Container Platform.

However, the NSX DHCP service is used for virtual machine IP management with the full-stack automated OpenShift Container Platform deployment and with nodes provisioned, either manually or automatically, by the Machine API integration with vSphere. Additionally, NSX firewall rules are created to enable access with the OpenShift Container Platform cluster and between the bastion host and the VMC vSphere hosts.

6.1.1. VMC Sizer tool

VMware Cloud on AWS is built on top of AWS bare metal infrastructure; this is the same bare metal infrastructure which runs AWS native services. When a VMware cloud on AWS software-defined data center (SDDC) is deployed, you consume these physical server nodes and run the VMware ESXi hypervisor in a single tenant fashion. This means the physical infrastructure is not accessible to anyone else using VMC. It is important to consider how many physical hosts you will need to host your virtual infrastructure.

To determine this, VMware provides the VMC on AWS Sizer. With this tool, you can define the resources you intend to host on VMC:

- Types of workloads
- Total number of virtual machines
- Specification information such as:
  - Storage requirements
  - vCPUs
  - vRAM
  - Overcommit ratios

With these details, the sizer tool can generate a report, based on VMware best practices, and recommend your cluster configuration and the number of hosts you will need.

6.2. VSPHERE PREREQUISITES

- 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 block registry storage. For more information on persistent storage, see Understanding persistent storage.
- 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.

6.3. INTERNET ACCESS FOR OPENSHIFT CONTAINER PLATFORM

In OpenShift Container Platform 4.12, you require access to the internet to install your cluster.

You must have internet access to:

- Access OpenShift Cluster Manager Hybrid Cloud Console 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.

6.4. 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 7.0 Update 2 or later
- Version 8.0 Update 1 or later

You can host the VMware vSphere infrastructure on-premise or on a VMware Cloud Verified provider that meets the requirements outlined in the following table:

Table 6.1. Version requirements for vSphere virtual environments

Virtual environment product	Required version
VMware virtual hardware	15 or later
vSphere ESXi hosts	7.0 Update 2 or later; 8.0 Update 1 or later
vCenter host	7.0 Update 2 or later; 8.0 Update 1 or later

IMPORTANT

Installing a cluster on VMware vSphere versions 7.0 and 7.0 Update 1 is deprecated. These versions are still fully supported, but all vSphere 6.x versions are no longer supported. Version 4.12 of OpenShift Container Platform requires VMware virtual hardware version 15 or later. To update the hardware version for your vSphere virtual machines, see the "Updating hardware on nodes running in vSphere" article in the *Updating clusters* section.

Table 6.2. Minimum supported vSphere version for VMware components

Component	Minimum supported versions	Description

Component	Minimum supported versions	Description
Hypervisor	vSphere 7.0 Update 2 (or later) or vSphere 8.0 Update 1 (or later) with virtual hardware version 15	This 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.
Storage with in-tree drivers	vSphere 7.0 Update 2 or later; 8.0 Update 1 or later	This plugin creates vSphere storage by using the in-tree storage drivers for vSphere included in OpenShift Container Platform.

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.

6.5. VMWARE VSPHERE CSI DRIVER OPERATOR REQUIREMENTS

To install the vSphere CSI Driver Operator, the following requirements must be met:

- VMware vSphere version: 7.0 Update 2 or later; 8.0 Update 1 or later
- vCenter version: 7.0 Update 2 or later; 8.0 Update 1 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.

Additional resources

- To remove a third-party 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.

6.6. 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.

6.6.1. vCenter requirements

Before you install an OpenShift Container Platform cluster on your vCenter that uses infrastructure that you provided, you must prepare your environment.

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 6.1. 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 will be created 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	If an existing resource pool is provided	**Host.Config.Storage 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 VirtualMachine.Config.ResetGuestInfo VirtualMachine.Config.Resource VirtualMachine.Config.Setti**

vSphere object for role	When required	Required privileges in vSphere API
		ngs 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 Datacenter	If the installation program creates the virtual machine folder. For UPI, **VirtualMachine.Inventory.Create** and **VirtualMachine.Inventory.Delete** privileges are optional if your cluster does not use the Machine API.	**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** **VirtualMachine.Config.ResetGuestInfo** **VirtualMachine.Config.Resource**

vSphere object for role	When required	Required privileges in vSphere API
		VirtualMachine.Config.Settings 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.DeployTemplate VirtualMachine.Provisioning.MarkAsTemplate Folder.Create Folder.Delete

Example 6.2. 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" "Profile-driven storage"."Profile-driven storage update" "Profile-driven storage"."Profile-driven storage view"**
vSphere vCenter Cluster	If VMs will be created 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 an existing resource pool is provided	**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 object for role	When required	Required privileges in vCenter GUI
vSphere Datastore	Always	**Datastore."Allocate space" Datastore."Browse datastore" Datastore."Low level file operations" "vSphere Tagging"."Assign or Unassign vSphere Tag on Object"**
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**

vSphere object for role	When required	Required privileges in vCenter GUI
		Configuration".Rename "Virtual machine"."Change Configuration"."Reset guest information" "Virtual machine"."Change Configuration"."Change resource" "Virtual machine"."Change Configuration"."Change Settings" "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 Datacenter	If the installation program creates the virtual machine folder. For UPI, **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

vSphere object for role	When required	Required privileges in vCenter GUI
		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 Settings" "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

vSphere object for role	When required	Required privileges in vCenter GUI
		Inventory"."Create from existing "Virtual machine"."Edit Inventory"."Remove" "Virtual machine".Provisioning."Clone virtual machine" "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 propogate the permissions to child objects. These settings vary depending on whether or not you install the cluster into an existing folder.

Example 6.3. Required permissions and propagation settings

vSphere object	When required	Propagate to children	Permissions required
vSphere vCenter	Always	False	Listed required privileges
vSphere vCenter Datacenter	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

vSphere object	When required	Propagate to children	Permissions required

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

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.

- OpenShift Container Platform generally supports compute-only vMotion, 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.

- Using Storage vMotion can cause issues and is not supported. If you are using 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 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.

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 is destroyed during the cluster installation process. A minimum of 800 GB of storage is required to use a standard cluster.

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

Cluster limits

Available resources vary between clusters. The number of possible clusters within a vCenter is limited 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.

Networking requirements

You must use the Dynamic Host Configuration Protocol (DHCP) for the network and ensure that the DHCP server is configured to provide persistent IP addresses to the cluster machines. In the DHCP lease, you must configure the DHCP to use the default gateway. All nodes must be in the same VLAN. You cannot scale the cluster using a second VLAN as a Day 2 operation. Additionally, you must create the following networking resources before you install the OpenShift Container Platform cluster:

NOTE

It is recommended that each OpenShift Container Platform node in the cluster must have access to a Network Time Protocol (NTP) server that is discoverable via DHCP. Installation is possible without an NTP server. However, asynchronous server clocks will cause errors, which NTP server prevents.

Required IP Addresses

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 6.3. Required DNS records

Component	Record	Description
API VIP	**api.<cluster_name>.<base_domain>.**	This DNS A/AAAA or CNAME 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	***.apps.<cluster_name>.<base_domain>.**	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

6.6.2. Required machines for cluster installation

The smallest OpenShift Container Platform clusters require the following hosts:

Table 6.4. 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.
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) 8 and inherits all of its hardware certifications and requirements. See Red Hat Enterprise Linux technology capabilities and limits .

6.6.3. Minimum resource requirements for cluster installation

Each cluster machine must meet the following minimum requirements:

Table 6.5. Minimum resource requirements

Machine	Operating System	vCPU [1]	Virtual RAM	Storage	Input/Output Per Second (IOPS)[2]
Bootstrap	RHCOS	4	16 GB	100 GB	300
Control plane	RHCOS	4	16 GB	100 GB	300
Compute	RHCOS, RHEL 8.6 and later [3]	2	8 GB	100 GB	300

Machine	Operating System	vCPU [1]	Virtual RAM	Storage	Input/Output Per Second (IOPS)[2]

1. One vCPU is equivalent to one physical core when simultaneous multithreading (SMT), or hyperthreading, is not enabled. When enabled, use the following formula to calculate the corresponding ratio: (threads per core × cores) × sockets = vCPUs.

2. 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.

3. 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.

If an instance type for your platform meets the minimum requirements for cluster machines, it is supported to use in OpenShift Container Platform.

6.6.4. Certificate signing requests management

Because your cluster has limited access to automatic machine management when you use infrastructure that you provision, you must provide a mechanism for approving cluster certificate signing requests (CSRs) after installation. 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.

6.6.5. Networking requirements for user-provisioned infrastructure

All the Red Hat Enterprise Linux CoreOS (RHCOS) machines require networking to be configured in **initramfs** during boot to fetch their Ignition config files.

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.

It is recommended to use 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.

NOTE

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.

6.6.5.1. 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.

6.6.5.2. 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 6.6. 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

Protocol	Port	Description
	10256	openshift-sdn
UDP	**4789**	VXLAN
	6081	Geneve
	9000-9999	Host level services, including the node exporter on ports **9100-9101**.
	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	IPsec Encapsulating Security Payload (ESP)

Table 6.7. Ports used for all-machine to control plane communications

Protocol	Port	Description
TCP	**6443**	Kubernetes API

Table 6.8. Ports used for control plane machine to control plane machine communications

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

Ethernet adaptor hardware address requirements

When provisioning VMs for the cluster, the ethernet interfaces configured for each VM must use a MAC address from the VMware Organizationally Unique Identifier (OUI) allocation ranges:

- **00:05:69:00:00:00** to **00:05:69:FF:FF:FF**
- **00:0c:29:00:00:00** to **00:0c:29:FF:FF:FF**
- **00:1c:14:00:00:00** to **00:1c:14:FF:FF:FF**
- **00:50:56:00:00:00** to **00:50:56:3F:FF:FF**

If a MAC address outside the VMware OUI is used, the cluster installation will not succeed.

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.

6.6.6. User-provisioned DNS requirements

In OpenShift Container Platform deployments, DNS name resolution is required for the following 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 6.9. Required DNS records

Component	Record	Description
Kubernetes API	**api.<cluster_name>.<base_domain>.**	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.

Component	Record	Description
	api-int.<cluster_name>.<base_domain>.	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.
Routes	***.apps.<cluster_name>.<base_domain>.**	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, **console-openshift-console.apps.<cluster_name>.<base_domain>** is used as a wildcard route to the OpenShift Container Platform console.
Bootstrap machine	**bootstrap.<cluster_name>.<base_domain>.**	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	**<control_plane><n>.<cluster_name>.<base_domain>.**	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	**<compute><n>.<cluster_name>.<base_domain>.**	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.

6.6.6.1. Example DNS configuration for user-provisioned clusters

This section provides A and PTR record configuration samples that meet the DNS requirements for deploying OpenShift Container Platform on user-provisioned infrastructure. The samples 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**.

Example DNS A record configuration for a user-provisioned cluster

The following example is a BIND zone file that shows sample A records for name resolution in a user-provisioned cluster.

Example 6.4. Sample DNS zone database

```
$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 1
api-int.ocp4.example.com. IN A 192.168.1.5 2
;
*.apps.ocp4.example.com. IN A 192.168.1.5 3
;
bootstrap.ocp4.example.com. IN A 192.168.1.96 4
;
control-plane0.ocp4.example.com. IN A 192.168.1.97 5
control-plane1.ocp4.example.com. IN A 192.168.1.98 6
control-plane2.ocp4.example.com. IN A 192.168.1.99 7
;
compute0.ocp4.example.com. IN A 192.168.1.11 8
compute1.ocp4.example.com. IN A 192.168.1.7 9
;
;EOF
```

1. Provides name resolution for the Kubernetes API. The record refers to the IP address of the API load balancer.

2. 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.

3. 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

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.

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.

4 Provides name resolution for the bootstrap machine.

5 6 7 Provides name resolution for the control plane machines.

8 9 Provides name resolution for the compute machines.

Example DNS PTR record configuration for a user-provisioned cluster

The following example BIND zone file shows sample PTR records for reverse name resolution in a user-provisioned cluster.

Example 6.5. Sample DNS zone database for reverse records

```
$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. 1
5.1.168.192.in-addr.arpa. IN PTR api-int.ocp4.example.com. 2
;
96.1.168.192.in-addr.arpa. IN PTR bootstrap.ocp4.example.com. 3
;
97.1.168.192.in-addr.arpa. IN PTR control-plane0.ocp4.example.com. 4
98.1.168.192.in-addr.arpa. IN PTR control-plane1.ocp4.example.com. 5
99.1.168.192.in-addr.arpa. IN PTR control-plane2.ocp4.example.com. 6
;
11.1.168.192.in-addr.arpa. IN PTR compute0.ocp4.example.com. 7
7.1.168.192.in-addr.arpa. IN PTR compute1.ocp4.example.com. 8
;
;EOF
```

1 Provides reverse DNS resolution for the Kubernetes API. The PTR record refers to the record name of the API load balancer.

2 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.

3 Provides reverse DNS resolution for the bootstrap machine.

4 5 6 Provides reverse DNS resolution for the control plane machines.

7 8 Provides reverse DNS resolution for the compute machines.

NOTE

A PTR record is not required for the OpenShift Container Platform application wildcard.

6.6.7. 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:

1. **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 load balancers:

   Table 6.10. API load balancer

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

Port	Back-end machines (pool members)	Internal	External	Description
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.

2. **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 6.11. 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.

6.6.7.1. Example load balancer configuration for user-provisioned clusters

This section provides an example API and application ingress load balancer configuration that meets 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.

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**.

Example 6.6. 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 (1)
  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 (2)
```

```
  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 3
  bind *:22623
  mode tcp
  server bootstrap bootstrap.ocp4.example.com:22623 check inter 1s backup 4
  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 5
  bind *:443
  mode tcp
  balance source
  server worker0 worker0.ocp4.example.com:443 check inter 1s
  server worker1 worker1.ocp4.example.com:443 check inter 1s
listen ingress-router-80 6
  bind *:80
  mode tcp
  balance source
  server worker0 worker0.ocp4.example.com:80 check inter 1s
  server worker1 worker1.ocp4.example.com:80 check inter 1s
```

1 Port **6443** handles the Kubernetes API traffic and points to the control plane machines.

2 4 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.

3 Port **22623** handles the machine config server traffic and points to the control plane machines.

5 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.

6 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.

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 -nltupe** on the HAProxy node.

6.7. PREPARING THE USER-PROVISIONED INFRASTRUCTURE

Before you install OpenShift Container Platform on user-provisioned infrastructure, you must prepare the underlying infrastructure.

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

1. If you are using DHCP to provide the IP networking configuration to your cluster nodes, configure your DHCP service.

   a. 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.

6.8. VALIDATING DNS RESOLUTION FOR USER-PROVISIONED INFRASTRUCTURE

You can validate your DNS configuration before installing OpenShift Container Platform on user-provisioned infrastructure.

> **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> 1
   ```

   1 Replace **<nameserver_ip>** with the IP address of the nameserver, **<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
   ```

   c. 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
```

d. 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
```

e. 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.

2. 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.

a. 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. 1
5.1.168.192.in-addr.arpa. 604800 IN PTR api.ocp4.example.com. 2
```

1 Provides the record name for the Kubernetes internal API.

2 Provides 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.

6.9. GENERATING A KEY PAIR FOR CLUSTER NODE SSH ACCESS

During an OpenShift Container Platform installation, you can provide an SSH public key to the installation program. The key is passed to the Red Hat Enterprise Linux CoreOS (RHCOS) nodes through their Ignition config files and is used to authenticate SSH access to the nodes. The key is added to the **~/.ssh/authorized_keys** list for the **core** user on each node, which enables password-less authentication.

After the key is passed to the nodes, 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 such as AWS key pairs.

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> 1
```

1 Specify 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 FIPS validated or Modules In Process cryptographic libraries on 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> 1
   ```

   1 Specify 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.

6.10. VMWARE VSPHERE REGION AND ZONE ENABLEMENT

You can deploy an OpenShift Container Platform cluster to multiple vSphere datacenters that run in a single VMware vCenter. Each datacenter 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

VMware vSphere region and zone enablement 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 .

The default installation configuration deploys a cluster to a single vSphere datacenter. If you want to deploy a cluster to multiple vSphere datacenters, 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 datacenters 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 datacenter.

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

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

Table 6.12. Example of a configuration with multiple vSphere datacenters that run in a single VMware vCenter

Datacenter (region)	Cluster (zone)	Tags
us-east	us-east-1	us-east-1a

Datacenter (region)	Cluster (zone)	Tags
		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

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

2. Select your infrastructure provider.

3. Navigate to the page for your installation type, download the installation program that corresponds with your host operating system and architecture, and place the file in the directory where you will store the installation configuration files.

   IMPORTANT

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

   IMPORTANT

   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.

4. 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
```

5. Download your installation pull secret from the 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.

6.12. MANUALLY CREATING THE INSTALLATION CONFIGURATION FILE

Installing the cluster requires that you manually create the installation configuration file.

Prerequisites

- You have an SSH public key on your local machine to provide to the installation program. The key will be used 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, 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.

2. Customize the sample **install-config.yaml** file template that is provided and save it 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 multiple clusters.

IMPORTANT

The **install-config.yaml** file is consumed during the next step of the installation process. You must back it up now.

6.12.1. 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.

```
apiVersion: v1
baseDomain: example.com 1
compute: 2
  name: worker
  replicas: 0 3
controlPlane: 4
  name: master
  replicas: 3 5
metadata:
  name: test 6
platform:
  vsphere:
    vcenter: your.vcenter.server 7
    username: username 8
    password: password 9
    datacenter: datacenter 10
    defaultDatastore: datastore 11
    folder: "/<datacenter_name>/vm/<folder_name>/<subfolder_name>" 12
    resourcePool: "/<datacenter_name>/host/<cluster_name>/Resources/<resource_pool_name>" 13
    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. Although both sections currently define a single machine pool, it is possible that future versions of OpenShift Container Platform will support defining multiple compute pools during installation. Only one control plane pool is used.

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 The fully-qualified hostname or IP address of the vCenter server.

IMPORTANT

The Cluster 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.

8. The name of the user for accessing the server.

9. The password associated with the vSphere user.

10. The vSphere datacenter.

11. The default vSphere datastore to use.

12. Optional parameter: For installer-provisioned infrastructure, the absolute path of an existing folder where the installation program creates the virtual machines, for example, **/<datacenter_name>/vm/<folder_name>/<subfolder_name>**. If you do not provide this value, the installation program creates a top-level folder in the datacenter 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.

13. Optional parameter: For installer-provisioned infrastructure, the absolute path of an existing folder where the installation program creates the virtual machines, for example, **/<datacenter_name>/vm/<folder_name>/<subfolder_name>**. If you do not provide this value, the installation program creates a top-level folder in the datacenter virtual machine folder that is named with the infrastructure ID. If you are providing the infrastructure for the cluster, omit this parameter.

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 Installing the system in FIPS mode. The use of FIPS validated or Modules In Process cryptographic libraries is only supported on OpenShift Container Platform deployments on the **x86_64**, **ppc64le**, and **s390x** architectures.

16. The pull secret that you obtained from OpenShift Cluster Manager Hybrid Cloud Console . 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).

6.12.2. Configuring the cluster-wide proxy during installation

Production environments can deny direct access to the internet and instead have an HTTP or HTTPS proxy available. You can configure a new OpenShift Container Platform cluster to use a proxy by configuring the proxy settings in the **install-config.yaml** file.

Prerequisites

- You have an existing **install-config.yaml** file.
- You 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 Platform (GCP), 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> 1
  httpsProxy: https://<username>:<pswd>@<ip>:<port> 2
  noProxy: example.com 3
additionalTrustBundle: | 4
    -----BEGIN CERTIFICATE-----
    <MY_TRUSTED_CA_CERT>
    -----END CERTIFICATE-----
additionalTrustBundlePolicy: <policy_to_add_additionalTrustBundle> 5
```

1. A proxy URL to use for creating HTTP connections outside the cluster. The URL scheme must be **http**.
2. A proxy URL to use for creating HTTPS connections outside the cluster.
3. 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.
4. If provided, the installation program generates a config map that is named **user-ca-bundle** in the **openshift-config** namespace that contains one or more additional CA certificates

that are required for proxying HTTPS connections. The Cluster Network Operator then creates a **trusted-ca-bundle** config map that merges these contents with the Red Hat Enterprise Linux CoreOS (RHCOS) trust bundle, and this config map is referenced in the **trustedCA** field of the **Proxy** object. The **additionalTrustBundle** field is required unless the proxy's identity certificate is signed by an authority from the RHCOS trust bundle.

5 Optional: The policy to determine 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**.

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.

6.12.3. Configuring regions and zones for a VMware vCenter

You can modify the default installation configuration file to deploy an OpenShift Container Platform cluster to multiple vSphere datacenters that run in a single VMware vCenter.

IMPORTANT

VMware vSphere region and zone enablement 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 .

IMPORTANT

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

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 datacenter objects for your VMware vCenter server. Consider specifying multiple failure domains if you need to provision virtual machine nodes in different datacenters, clusters, datastores, and other components. To enable regions and zones, you must define multiple failure domains for your OpenShift Container Platform cluster.

  NOTE

  You cannot change a failure domain after you installed an OpenShift Container Platform cluster on the VMware vSphere platform. You can add additional failure domains after cluster installation.

Procedure

1. Enter the following **govc** command-line tool commands to create the **openshift-region** and **openshift-zone** vCenter tag categories:

   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. To create a region tag for each region vSphere datacenter where you want to deploy your cluster, enter the following command in your terminal:

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

3. To create a zone tag for each vSphere cluster where you want to deploy your cluster, enter the following command:

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

4. Attach region tags to each vCenter datacenter object by entering the following command:

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

5. Attach the zone tags to each vCenter datacenter object by entering the following command:

```
$ govc tags.attach -c <zone_tag_category> <zone_tag_1> /<datacenter_1>/host/vcs-mdcnc-workload-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 datacenters defined in a vSphere center

```
apiVersion: v1
baseDomain: example.com
featureSet: TechPreviewNoUpgrade 1
compute:
  name: worker
  replicas: 3
  vsphere:
    zones: 2
      - "<machine_pool_zone_1>"
      - "<machine_pool_zone_2>"
controlPlane:
  name: master
  replicas: 3
  vsphere:
    zones: 3
      - "<machine_pool_zone_1>"
      - "<machine_pool_zone_2>"
metadata:
  name: cluster
platform:
  vsphere:
    vcenter: <vcenter_server> 4
    username: <username> 5
    password: <password> 6
    datacenter: datacenter 7
    defaultDatastore: datastore 8
    folder: "/<datacenter_name>/vm/<folder_name>/<subfolder_name>" 9
    cluster: cluster 10
    resourcePool: "/<datacenter_name>/host/<cluster_name>/Resources/<resource_pool_name>" 11
    diskType: thin
    failureDomains: 12
    - name: <machine_pool_zone_1> 13
      region: <region_tag_1> 14
      zone: <zone_tag_1> 15
      topology: 16
        datacenter: <datacenter1> 17
        computeCluster: "/<datacenter1>/host/<cluster1>" 18
        resourcePool: "/<datacenter1>/host/<cluster1>/Resources/<resourcePool1>" 19
        networks: 20
        - <VM_Network1_name>
```

```
      datastore: "/<datacenter1>/datastore/<datastore1>" 21
  - name: <machine_pool_zone_2>
    region: <region_tag_2>
    zone: <zone_tag_2>
    topology:
      datacenter: <datacenter2>
      computeCluster: "/<datacenter2>/host/<cluster2>"
      networks:
      - <VM_Network2_name>
      datastore: "/<datacenter2>/datastore/<datastore2>"
      resourcePool: "/<datacenter2>/host/<cluster2>/Resources/<resourcePool2>"
      folder: "/<datacenter2>/vm/<folder2>"
# ...
```

1 You must define set the **TechPreviewNoUpgrade** as the value for this parameter, so that you can use the VMware vSphere region and zone enablement feature.

2 3 An optional parameter for specifying a vCenter cluster. You define a zone by using a tag from the **openshift-zone** tag category. If you do not define this parameter, nodes will be distributed among all defined failure-domains.

4 5 6 7 8 9 10 11 The default vCenter topology. The installation program uses this topology information to deploy the bootstrap node. Additionally, the topology defines the default datastore for vSphere persistent volumes.

12 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. If you do not define this parameter, the installation program uses the default vCenter topology.

13 Defines the name of the failure domain. Each failure domain is referenced in the **zones** parameter to scope a machine pool to the failure domain.

14 You define a region by using a tag from the **openshift-region** tag category. The tag must be attached to the vCenter datacenter.

15 You define a zone by using a tag from the **openshift-zone tag** category. The tag must be attached to the vCenter datacenter.

16 Specifies the vCenter resources associated with the failure domain.

17 An optional parameter for defining the vSphere datacenter that is associated with a failure domain. If you do not define this parameter, the installation program uses the default vCenter topology.

18 An optional parameter for stating the absolute file path for the compute cluster that is associated with the failure domain. If you do not define this parameter, the installation program uses the default vCenter topology.

19 An optional parameter for the installer-provisioned infrastructure. The parameter sets the absolute path of an existing resource pool where the installation program creates the virtual machines, for example, **/<datacenter_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_datacenter/host/example_cluster/Resources**.

20 An optional parameter that lists any network in the vCenter instance that contains the virtual IP addresses and DNS records that you configured. If you do not define this parameter, the

installation program uses the default vCenter topology.

21 An optional parameter for specifying a datastore to use for provisioning volumes. If you do not define this parameter, the installation program uses the default vCenter topology.

6.13. CREATING THE KUBERNETES MANIFEST AND IGNITION CONFIG FILES

Because you must modify some cluster definition files and manually start the cluster machines, you must generate the Kubernetes manifest and Ignition config files that the cluster needs to configure the machines.

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> 1
```

1 For **<installation_directory>**, specify the installation directory that contains the **install-config.yaml** file you created.

2. Remove the Kubernetes manifest files that define the control plane machines and compute machine sets:

```
$ 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 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.yml** 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.yml** 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> 1
```

1 For **<installation_directory>**, specify 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
```

6.14. EXTRACTING THE INFRASTRUCTURE NAME

The Ignition config files contain a unique cluster identifier that you can use to uniquely identify your cluster in VMware Cloud on AWS. 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.

6.15. 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 datacenter to store your VMs.

a. Click the **VMs and Templates** view.

b. Right-click the name of your datacenter.

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.

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 datacenter.

    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.

6.16. 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 datacenter.
5. On the **Select storage** tab, select storage for your configuration and disk files.
6. On the **Select clone options**, select **Customize this virtual machine's hardware**.
7. On the **Customize hardware** tab, click **Advanced**.
   a. Click **Edit Configuration**, and on the **Configuration Parameters** window, click **Add Configuration Params**. Define the following parameter names and values:
      - **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.

6.17. 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.

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: 4.12.0
metadata:
  labels:
    machineconfiguration.openshift.io/role: worker
  name: 98-var-partition
storage:
  disks:
  - device: /dev/<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
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.

6.18. INSTALLING THE OPENSHIFT CLI BY DOWNLOADING THE BINARY

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

> **IMPORTANT**
>
> If you installed an earlier version of **oc**, you cannot use it to complete all of the commands in OpenShift Container Platform 4.12. Download and install the new version of **oc**.

Installing the OpenShift CLI on Linux

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

Procedure

1. Navigate to the OpenShift Container Platform downloads page on the Red Hat Customer Portal.
2. Select the architecture from the **Product Variant** drop-down list.
3. Select the appropriate version from the **Version** drop-down list.

4. Click **Download Now** next to the **OpenShift v4.12 Linux Client** 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>
```

Installing the OpenShift CLI on Windows

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

Procedure

1. Navigate to the OpenShift Container Platform downloads page on the Red Hat Customer Portal.

2. Select the appropriate version from the **Version** drop-down list.

3. Click **Download Now** next to the **OpenShift v4.12 Windows Client** entry and save the file.

4. Unzip the archive with a ZIP program.

5. Move the **oc** binary to a directory that is on your **PATH**.
   To check your **PATH**, open the command prompt and execute the following command:

```
C:\> path
```

Verification

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

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

Installing the OpenShift CLI on macOS

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

Procedure

1. Navigate to the OpenShift Container Platform downloads page on the Red Hat Customer Portal.

2. Select the appropriate version from the **Version** drop-down list.

3. Click **Download Now** next to the **OpenShift v4.12 macOS Client** entry and save the file.

> **NOTE**
>
> For macOS arm64, choose the **OpenShift v4.12 macOS arm64 Client** entry.

4. Unpack and unzip the archive.

5. Move the **oc** binary to a directory on your PATH.
   To check your **PATH**, open a terminal and execute the following command:

   ```
   $ echo $PATH
   ```

Verification

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

  ```
  $ oc <command>
  ```

6.19. WAITING FOR THE BOOTSTRAP PROCESS TO COMPLETE

The OpenShift Container Platform bootstrap process begins after the cluster nodes first boot into the persistent RHCOS environment that has been installed to disk. The configuration information provided through the Ignition config files is used to initialize the bootstrap process and install OpenShift Container Platform on the machines. You must wait for the bootstrap process to complete.

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 \ 1
       --log-level=info 2
   ```

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

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

   #### Example output

```
INFO Waiting up to 30m0s for the Kubernetes API at https://api.test.example.com:6443...
INFO API v1.25.0 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.

6.20. LOGGING IN TO THE CLUSTER BY USING THE CLI

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

Prerequisites

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

Procedure

1. Export the **kubeadmin** credentials:

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

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

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

```
$ oc whoami
```

Example output

```
system:admin
```

6.21. APPROVING THE CERTIFICATE SIGNING REQUESTS FOR YOUR MACHINES

When you add machines to a cluster, two pending certificate signing requests (CSRs) are generated for each machine that you added. You must confirm that these CSRs are approved or, if necessary, approve them yourself. The client requests must be approved 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  v1.25.0
master-1  Ready     master  63m  v1.25.0
master-2  Ready     master  64m  v1.25.0
```

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> 1
```

1 **<csr_name>** is 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> 1
     ```

     1 **<csr_name>** is 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  v1.25.0
   master-1  Ready     master  73m  v1.25.0
   master-2  Ready     master  74m  v1.25.0
   worker-0  Ready     worker  11m  v1.25.0
   worker-1  Ready     worker  11m  v1.25.0
   ```

   NOTE

   It can take a few minutes after approval of the server CSRs for the machines to transition to the **Ready** status.

Additional information

- For more information on CSRs, see Certificate Signing Requests.

6.22. INITIAL OPERATOR CONFIGURATION

After the control plane initializes, you must immediately configure some Operators so that they all become available.

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                               4.12.0    True        False         False      19m
baremetal                                    4.12.0    True        False         False      37m
cloud-credential                             4.12.0    True        False         False      40m
cluster-autoscaler                           4.12.0    True        False         False      37m
config-operator                              4.12.0    True        False         False      38m
console                                      4.12.0    True        False         False      26m
csi-snapshot-controller                      4.12.0    True        False         False      37m
dns                                          4.12.0    True        False         False      37m
etcd                                         4.12.0    True        False         False      36m
image-registry                               4.12.0    True        False         False      31m
ingress                                      4.12.0    True        False         False      30m
insights                                     4.12.0    True        False         False      31m
kube-apiserver                               4.12.0    True        False         False      26m
kube-controller-manager                      4.12.0    True        False         False      36m
kube-scheduler                               4.12.0    True        False         False      36m
kube-storage-version-migrator                4.12.0    True        False         False      37m
machine-api                                  4.12.0    True        False         False      29m
machine-approver                             4.12.0    True        False         False      37m
machine-config                               4.12.0    True        False         False      36m
marketplace                                  4.12.0    True        False         False      37m
monitoring                                   4.12.0    True        False         False      29m
network                                      4.12.0    True        False         False      38m
node-tuning                                  4.12.0    True        False         False      37m
openshift-apiserver                          4.12.0    True        False         False      32m
openshift-controller-manager                 4.12.0    True        False         False      30m
openshift-samples                            4.12.0    True        False         False      32m
operator-lifecycle-manager                   4.12.0    True        False         False      37m
operator-lifecycle-manager-catalog           4.12.0    True        False         False      37m
operator-lifecycle-manager-packageserver     4.12.0    True        False         False      32m
service-ca                                   4.12.0    True        False         False      38m
storage                                      4.12.0    True        False         False      37m
```

2. Configure the Operators that are not available.

6.22.1. 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.

6.22.2. 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.

Instructions are shown for configuring a persistent volume, which is required for production clusters. Where applicable, instructions are shown for configuring an empty directory as the storage location, which is available for only non-production clusters.

Additional instructions are provided for allowing the image registry to use block storage types by using the **Recreate** rollout strategy during upgrades.

6.22.2.1. 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. To configure your registry to use storage, change the **spec.storage.pvc** 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:

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

Example output

```
No resourses 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:

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

Example output

```
storage:
  pvc:
    claim: 1
```

1. 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:

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

Example output

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

6.22.2.2. 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 for only 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.

6.22.2.3. 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 PV for the block storage device, and create a 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 1
  namespace: openshift-image-registry 2
spec:
  accessModes:
  - ReadWriteOnce 3
  resources:
    requests:
      storage: 100Gi 4
```

1 A unique name that represents the **PersistentVolumeClaim** object.

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

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

4. 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: 1
```

1. 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.

6.23. COMPLETING INSTALLATION ON USER-PROVISIONED INFRASTRUCTURE

After you complete the Operator configuration, you can finish installing the cluster on 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                             4.12.0    True       False       False      19m
baremetal                                  4.12.0    True       False       False      37m
cloud-credential                           4.12.0    True       False       False      40m
cluster-autoscaler                         4.12.0    True       False       False      37m
config-operator                            4.12.0    True       False       False      38m
console                                    4.12.0    True       False       False      26m
csi-snapshot-controller                    4.12.0    True       False       False      37m
dns                                        4.12.0    True       False       False      37m
etcd                                       4.12.0    True       False       False      36m
image-registry                             4.12.0    True       False       False      31m
ingress                                    4.12.0    True       False       False      30m
insights                                   4.12.0    True       False       False      31m
kube-apiserver                             4.12.0    True       False       False      26m
kube-controller-manager                    4.12.0    True       False       False      36m
kube-scheduler                             4.12.0    True       False       False      36m
kube-storage-version-migrator              4.12.0    True       False       False      37m
machine-api                                4.12.0    True       False       False      29m
machine-approver                           4.12.0    True       False       False      37m
machine-config                             4.12.0    True       False       False      36m
marketplace                                4.12.0    True       False       False      37m
monitoring                                 4.12.0    True       False       False      29m
network                                    4.12.0    True       False       False      38m
node-tuning                                4.12.0    True       False       False      37m
openshift-apiserver                        4.12.0    True       False       False      32m
openshift-controller-manager               4.12.0    True       False       False      30m
openshift-samples                          4.12.0    True       False       False      32m
operator-lifecycle-manager                 4.12.0    True       False       False      37m
operator-lifecycle-manager-catalog         4.12.0    True       False       False      37m
operator-lifecycle-manager-packageserver   4.12.0    True       False       False      32m
service-ca                                 4.12.0    True       False       False      38m
storage                                    4.12.0    True       False       False      37m
```

Alternatively, the following command notifies you when all of the clusters are available. It also retrieves and displays credentials:

```
$ ./openshift-install --dir <installation_directory> wait-for install-complete 1
```

1 For **<installation_directory>**, specify 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> (1)
```

(1) Specify the pod name and namespace, as shown in the output of the previous 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 *Post-installation 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.

6.24. BACKING UP VMWARE VSPHERE VOLUMES

OpenShift Container Platform provisions new volumes as independent persistent disks to freely attach and detach the volume on any node in the cluster. As a consequence, it is not possible to back up volumes that use snapshots, or to restore volumes from snapshots. See Snapshot Limitations for more information.

Procedure

To create a backup of persistent volumes:

1. Stop the application that is using the persistent volume.
2. Clone the persistent volume.
3. Restart the application.
4. Create a backup of the cloned volume.
5. Delete the cloned volume.

6.25. TELEMETRY ACCESS FOR OPENSHIFT CONTAINER PLATFORM

In OpenShift Container Platform 4.12, the Telemetry service, which runs by default to provide metrics about cluster health and the success of updates, requires internet access. If your cluster is connected to the internet, Telemetry runs automatically, and your cluster is registered to OpenShift Cluster Manager Hybrid Cloud Console.

After you confirm that your OpenShift Cluster Manager Hybrid Cloud Console 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.

Additional resources

- See About remote health monitoring for more information about the Telemetry service

6.26. NEXT STEPS

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

CHAPTER 7. INSTALLING A CLUSTER ON VMC WITH USER-PROVISIONED INFRASTRUCTURE AND NETWORK CUSTOMIZATIONS

In OpenShift Container Platform version 4.12, you can install a cluster on your VMware vSphere instance using infrastructure you provision with customized network configuration options by deploying it to VMware Cloud (VMC) on AWS.

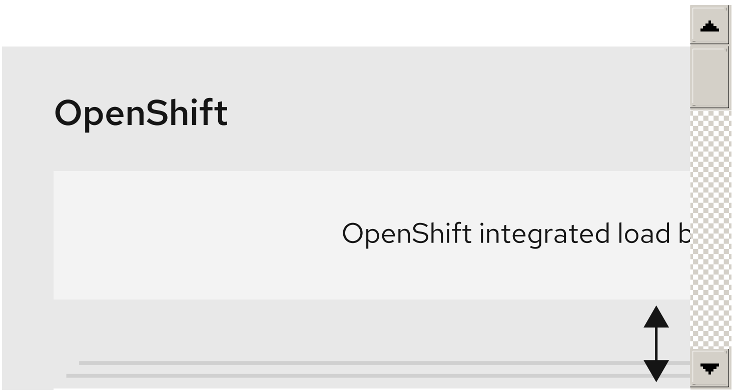
Once you configure your VMC environment for OpenShift Container Platform deployment, you use the OpenShift Container Platform installation program from the bastion management host, co-located in the VMC environment. The installation program and control plane automates the process of deploying and managing the resources needed for the OpenShift Container Platform cluster.

By customizing your network configuration, your cluster can coexist with existing IP address allocations in your environment and integrate with existing 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.

> **NOTE**
>
> OpenShift Container Platform supports deploying a cluster to a single VMware vCenter only. Deploying a cluster with machines/machine sets on multiple vCenters is not supported.

7.1. SETTING UP VMC FOR VSPHERE

You can install OpenShift Container Platform on VMware Cloud (VMC) on AWS hosted vSphere clusters to enable applications to be deployed and managed both on-premise and off-premise, across the hybrid cloud.



You must configure several options in your VMC environment prior to installing OpenShift Container Platform on VMware vSphere. Ensure your VMC environment has the following prerequisites:

- Create a non-exclusive, DHCP-enabled, NSX-T network segment and subnet. Other virtual machines (VMs) can be hosted on the subnet, but at least eight IP addresses must be available for the OpenShift Container Platform deployment.
- Configure the following firewall rules:
  - An ANY:ANY firewall rule between the OpenShift Container Platform compute network and the internet. This is used by nodes and applications to download container images.

- An ANY:ANY firewall rule between the installation host and the software-defined data center (SDDC) management network on port 443. This allows you to upload the Red Hat Enterprise Linux CoreOS (RHCOS) OVA during deployment.
- An HTTPS firewall rule between the OpenShift Container Platform compute network and vCenter. This connection allows OpenShift Container Platform to communicate with vCenter for provisioning and managing nodes, persistent volume claims (PVCs), and other resources.

- You must have the following information to deploy OpenShift Container Platform:
  - The OpenShift Container Platform cluster name, such as **vmc-prod-1**.
  - The base DNS name, such as **companyname.com**.
  - If not using the default, the pod network CIDR and services network CIDR must be identified, which are set by default to **10.128.0.0/14** and **172.30.0.0/16**, respectively. These CIDRs are used for pod-to-pod and pod-to-service communication and are not accessible externally; however, they must not overlap with existing subnets in your organization.
  - The following vCenter information:
    - vCenter hostname, username, and password
    - Datacenter name, such as **SDDC-Datacenter**
    - Cluster name, such as **Cluster-1**
    - Network name
    - Datastore name, such as **WorkloadDatastore**

NOTE

It is recommended to move your vSphere cluster to the VMC **Compute-ResourcePool** resource pool after your cluster installation is finished.

- A Linux-based host deployed to VMC as a bastion.
  - The bastion host can be Red Hat Enterprise Linux (RHEL) or any another Linux-based host; it must have internet connectivity and the ability to upload an OVA to the ESXi hosts.
  - Download and install the OpenShift CLI tools to the bastion host.
    - The **openshift-install** installation program
    - The OpenShift CLI (**oc**) tool

NOTE

You cannot use the VMware NSX Container Plugin for Kubernetes (NCP), and NSX is not used as the OpenShift SDN. The version of NSX currently available with VMC is incompatible with the version of NCP certified with OpenShift Container Platform.

However, the NSX DHCP service is used for virtual machine IP management with the full-stack automated OpenShift Container Platform deployment and with nodes provisioned, either manually or automatically, by the Machine API integration with vSphere. Additionally, NSX firewall rules are created to enable access with the OpenShift Container Platform cluster and between the bastion host and the VMC vSphere hosts.

7.1.1. VMC Sizer tool

VMware Cloud on AWS is built on top of AWS bare metal infrastructure; this is the same bare metal infrastructure which runs AWS native services. When a VMware cloud on AWS software-defined data center (SDDC) is deployed, you consume these physical server nodes and run the VMware ESXi hypervisor in a single tenant fashion. This means the physical infrastructure is not accessible to anyone else using VMC. It is important to consider how many physical hosts you will need to host your virtual infrastructure.

To determine this, VMware provides the VMC on AWS Sizer. With this tool, you can define the resources you intend to host on VMC:

- Types of workloads
- Total number of virtual machines
- Specification information such as:
  - Storage requirements
  - vCPUs
  - vRAM
  - Overcommit ratios

With these details, the sizer tool can generate a report, based on VMware best practices, and recommend your cluster configuration and the number of hosts you will need.

7.2. VSPHERE PREREQUISITES

- 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 block registry storage. For more information on persistent storage, see Understanding persistent storage.
- If you use a firewall, you configured it to allow the sites that your cluster requires access to.

7.3. INTERNET ACCESS FOR OPENSHIFT CONTAINER PLATFORM

In OpenShift Container Platform 4.12, you require access to the internet to install your cluster.

You must have internet access to:

- Access OpenShift Cluster Manager Hybrid Cloud Console 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.

7.4. 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 7.0 Update 2 or later
- Version 8.0 Update 1 or later

You can host the VMware vSphere infrastructure on-premise or on a VMware Cloud Verified provider that meets the requirements outlined in the following table:

Table 7.1. Version requirements for vSphere virtual environments

Virtual environment product	Required version
VMware virtual hardware	15 or later
vSphere ESXi hosts	7.0 Update 2 or later; 8.0 Update 1 or later
vCenter host	7.0 Update 2 or later; 8.0 Update 1 or later

IMPORTANT

Installing a cluster on VMware vSphere versions 7.0 and 7.0 Update 1 is deprecated. These versions are still fully supported, but all vSphere 6.x versions are no longer supported. Version 4.12 of OpenShift Container Platform requires VMware virtual hardware version 15 or later. To update the hardware version for your vSphere virtual machines, see the "Updating hardware on nodes running in vSphere" article in the *Updating clusters* section.

Table 7.2. Minimum supported vSphere version for VMware components

Component	Minimum supported versions	Description
Hypervisor	vSphere 7.0 Update 2 (or later) or vSphere 8.0 Update 1 (or later) with virtual hardware version 15	This 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.
Storage with in-tree drivers	vSphere 7.0 Update 2 or later; 8.0 Update 1 or later	This plugin creates vSphere storage by using the in-tree storage drivers for vSphere included in OpenShift Container Platform.

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.

7.5. VMWARE VSPHERE CSI DRIVER OPERATOR REQUIREMENTS

To install the vSphere CSI Driver Operator, the following requirements must be met:

- VMware vSphere version: 7.0 Update 2 or later; 8.0 Update 1 or later
- vCenter version: 7.0 Update 2 or later; 8.0 Update 1 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.

Additional resources

- To remove a third-party 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.

7.6. 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.

7.6.1. vCenter requirements

Before you install an OpenShift Container Platform cluster on your vCenter that uses infrastructure that you provided, you must prepare your environment.

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 7.1. 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 will be created 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	If an existing resource pool is provided	**Host.Config.Storage 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 VirtualMachine.Config.ResetGuestInfo VirtualMachine.Config.Resource VirtualMachine.Config.Setti**

vSphere object for role	When required	Required privileges in vSphere API
		ngs **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 Datacenter	If the installation program creates the virtual machine folder. For UPI, **VirtualMachine.Inventory.Create** and **VirtualMachine.Inventory.Delete** privileges are optional if your cluster does not use the Machine API.	**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** **VirtualMachine.Config.ResetGuestInfo** **VirtualMachine.Config.Resource** **VirtualMachine.Config.Setti**

vSphere object for role	When required	Required privileges in vSphere API
		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 7.2. 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" "Profile-driven storage"."Profile-driven storage update" "Profile-driven storage"."Profile-driven storage view"**
vSphere vCenter Cluster	If VMs will be created 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 an existing resource pool is provided	**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 object for role	When required	Required privileges in vCenter GUI
vSphere Datastore	Always	**Datastore."Allocate space" Datastore."Browse datastore" Datastore."Low level file operations" "vSphere Tagging"."Assign or Unassign vSphere Tag on Object"**
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**

vSphere object for role	When required	Required privileges in vCenter GUI
		Configuration".Rename **"Virtual machine"."Change Configuration"."Reset guest information"** **"Virtual machine"."Change Configuration"."Change resource"** **"Virtual machine"."Change Configuration"."Change Settings"** **"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 Datacenter	If the installation program creates the virtual machine folder. For UPI, **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"**

vSphere object for role	When required	Required privileges in vCenter GUI
		"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 Settings" "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"

vSphere object for role	When required	Required privileges in vCenter GUI
		"Virtual machine"."Edit Inventory"."Create from existing" **"Virtual machine"."Edit Inventory"."Remove"** **"Virtual machine".Provisioning."Clone virtual machine"** **"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 propogate the permissions to child objects. These settings vary depending on whether or not you install the cluster into an existing folder.

Example 7.3. Required permissions and propagation settings

vSphere object	When required	Propagate to children	Permissions required
vSphere vCenter	Always	False	Listed required privileges
vSphere vCenter Datacenter	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 object	When required	Propagate to children	Permissions required
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.

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.

- OpenShift Container Platform generally supports compute-only vMotion, 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.

- Using Storage vMotion can cause issues and is not supported. If you are using 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 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.

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 is destroyed during the cluster installation process. A minimum of 800 GB of storage is required to use a standard cluster.

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

Cluster limits

Available resources vary between clusters. The number of possible clusters within a vCenter is limited 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.

Networking requirements

You must use the Dynamic Host Configuration Protocol (DHCP) for the network and ensure that the DHCP server is configured to provide persistent IP addresses to the cluster machines. In the DHCP lease, you must configure the DHCP to use the default gateway. All nodes must be in the same VLAN. You cannot scale the cluster using a second VLAN as a Day 2 operation. Additionally, you must create the following networking resources before you install the OpenShift Container Platform cluster:

> **NOTE**
>
> It is recommended that each OpenShift Container Platform node in the cluster must have access to a Network Time Protocol (NTP) server that is discoverable via DHCP. Installation is possible without an NTP server. However, asynchronous server clocks will cause errors, which NTP server prevents.

Required IP Addresses

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 7.3. Required DNS records

Component	Record	Description
API VIP	**api.<cluster_name>.<base_domain>.**	This DNS A/AAAA or CNAME 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.

Component	Record	Description
Ingress VIP	***.apps.<cluster_name>.<base_domain>.**	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

7.6.2. Required machines for cluster installation

The smallest OpenShift Container Platform clusters require the following hosts:

Table 7.4. 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.
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) 8 and inherits all of its hardware certifications and requirements. See Red Hat Enterprise Linux technology capabilities and limits .

7.6.3. Minimum resource requirements for cluster installation

Each cluster machine must meet the following minimum requirements:

Table 7.5. Minimum resource requirements

Machine	Operating System	vCPU [1]	Virtual RAM	Storage	Input/Output Per Second (IOPS)[2]
Bootstrap	RHCOS	4	16 GB	100 GB	300
Control plane	RHCOS	4	16 GB	100 GB	300
Compute	RHCOS, RHEL 8.6 and later [3]	2	8 GB	100 GB	300

1. One vCPU is equivalent to one physical core when simultaneous multithreading (SMT), or hyperthreading, is not enabled. When enabled, use the following formula to calculate the corresponding ratio: (threads per core × cores) × sockets = vCPUs.

2. 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.

3. 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.

If an instance type for your platform meets the minimum requirements for cluster machines, it is supported to use in OpenShift Container Platform.

7.6.4. Certificate signing requests management

Because your cluster has limited access to automatic machine management when you use infrastructure that you provision, you must provide a mechanism for approving cluster certificate signing requests (CSRs) after installation. 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.

7.6.5. Networking requirements for user-provisioned infrastructure

All the Red Hat Enterprise Linux CoreOS (RHCOS) machines require networking to be configured in **initramfs** during boot to fetch their Ignition config files.

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.

It is recommended to use 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.

NOTE

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.

7.6.5.1. 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.

7.6.5.2. 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 7.6. Ports used for all-machine to all-machine communications

Protocol	Port	Description
ICMP	N/A	Network reachability tests

Protocol	Port	Description
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
	10256	openshift-sdn
UDP	**4789**	VXLAN
	6081	Geneve
	9000-9999	Host level services, including the node exporter on ports **9100-9101**.
	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	IPsec Encapsulating Security Payload (ESP)

Table 7.7. Ports used for all-machine to control plane communications

Protocol	Port	Description
TCP	**6443**	Kubernetes API

Table 7.8. Ports used for control plane machine to control plane machine communications

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

Ethernet adaptor hardware address requirements

When provisioning VMs for the cluster, the ethernet interfaces configured for each VM must use a MAC address from the VMware Organizationally Unique Identifier (OUI) allocation ranges:

- **00:05:69:00:00:00** to **00:05:69:FF:FF:FF**
- **00:0c:29:00:00:00** to **00:0c:29:FF:FF:FF**
- **00:1c:14:00:00:00** to **00:1c:14:FF:FF:FF**
- **00:50:56:00:00:00** to **00:50:56:3F:FF:FF**

If a MAC address outside the VMware OUI is used, the cluster installation will not succeed.

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.

7.6.6. User-provisioned DNS requirements

In OpenShift Container Platform deployments, DNS name resolution is required for the following 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 7.9. Required DNS records

Component	Record	Description
Kubernetes API	**api.<cluster_name>.<base_domain>.**	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.
	api-int.<cluster_name>.<base_domain>.	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.
Routes	***.apps.<cluster_name>.<base_domain>.**	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, **console-openshift-console.apps.<cluster_name>.<base_domain>** is used as a wildcard route to the OpenShift Container Platform console.
Bootstrap machine	**bootstrap.<cluster_name>.<base_domain>.**	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	**<control_plane><n>.<cluster_name>.<base_domain>.**	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	**<compute><n>.<cluster_name>.<base_domain>.**	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.

7.6.6.1. Example DNS configuration for user-provisioned clusters

This section provides A and PTR record configuration samples that meet the DNS requirements for deploying OpenShift Container Platform on user-provisioned infrastructure. The samples 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**.

Example DNS A record configuration for a user-provisioned cluster

The following example is a BIND zone file that shows sample A records for name resolution in a user-provisioned cluster.

Example 7.4. Sample DNS zone database

```
$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 1
api-int.ocp4.example.com. IN A 192.168.1.5 2
;
*.apps.ocp4.example.com. IN A 192.168.1.5 3
;
bootstrap.ocp4.example.com. IN A 192.168.1.96 4
;
control-plane0.ocp4.example.com. IN A 192.168.1.97 5
control-plane1.ocp4.example.com. IN A 192.168.1.98 6
control-plane2.ocp4.example.com. IN A 192.168.1.99 7
;
compute0.ocp4.example.com. IN A 192.168.1.11 8
compute1.ocp4.example.com. IN A 192.168.1.7 9
;
;EOF
```

1. Provides name resolution for the Kubernetes API. The record refers to the IP address of the API load balancer.

2 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.

3 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. The Ingress Controller pods run on the compute machines by default.

> **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.

4 Provides name resolution for the bootstrap machine.

5 6 7 Provides name resolution for the control plane machines.

8 9 Provides name resolution for the compute machines.

Example DNS PTR record configuration for a user-provisioned cluster

The following example BIND zone file shows sample PTR records for reverse name resolution in a user-provisioned cluster.

Example 7.5. Sample DNS zone database for reverse records

```
$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. 1
5.1.168.192.in-addr.arpa. IN PTR api-int.ocp4.example.com. 2
;
96.1.168.192.in-addr.arpa. IN PTR bootstrap.ocp4.example.com. 3
;
97.1.168.192.in-addr.arpa. IN PTR control-plane0.ocp4.example.com. 4
98.1.168.192.in-addr.arpa. IN PTR control-plane1.ocp4.example.com. 5
99.1.168.192.in-addr.arpa. IN PTR control-plane2.ocp4.example.com. 6
;
11.1.168.192.in-addr.arpa. IN PTR compute0.ocp4.example.com. 7
7.1.168.192.in-addr.arpa. IN PTR compute1.ocp4.example.com. 8
;
;EOF
```

1 Provides reverse DNS resolution for the Kubernetes API. The PTR record refers to the record name of the API load balancer.

2 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.

3 Provides reverse DNS resolution for the bootstrap machine.

4 5 6 Provides reverse DNS resolution for the control plane machines.

7 8 Provides reverse DNS resolution for the compute machines.

NOTE

A PTR record is not required for the OpenShift Container Platform application wildcard.

7.6.7. 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:

1. **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 load balancers:

   Table 7.10. API load balancer

Port	Back-end machines (pool members)	Internal	External	Description

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.

2. **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 7.11. Application Ingress load balancer

Port	Back-end machines (pool members)	Internal	External	Description

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.

7.6.7.1. Example load balancer configuration for user-provisioned clusters

This section provides an example API and application ingress load balancer configuration that meets 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.

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**.

Example 7.6. 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 1
  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 2
  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 3
  bind *:22623
  mode tcp
  server bootstrap bootstrap.ocp4.example.com:22623 check inter 1s backup 4
  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 5
  bind *:443
  mode tcp
  balance source
  server worker0 worker0.ocp4.example.com:443 check inter 1s
  server worker1 worker1.ocp4.example.com:443 check inter 1s
listen ingress-router-80 6
  bind *:80
  mode tcp
  balance source
  server worker0 worker0.ocp4.example.com:80 check inter 1s
  server worker1 worker1.ocp4.example.com:80 check inter 1s
```

1. Port **6443** handles the Kubernetes API traffic and points to the control plane machines.

2, 4. 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.

3. Port **22623** handles the machine config server traffic and points to the control plane machines.

5. 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.

6. 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.

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 -nltupe** on the HAProxy node.

7.7. PREPARING THE USER-PROVISIONED INFRASTRUCTURE

Before you install OpenShift Container Platform on user-provisioned infrastructure, you must prepare the underlying infrastructure.

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

1. If you are using DHCP to provide the IP networking configuration to your cluster nodes, configure your DHCP service.
   a. 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.

7.8. VALIDATING DNS RESOLUTION FOR USER-PROVISIONED INFRASTRUCTURE

You can validate your DNS configuration before installing OpenShift Container Platform on user-provisioned infrastructure.

> **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> 1
   ```

   1 Replace **<nameserver_ip>** with the IP address of the nameserver, **<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
   ```

c. 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
```

d. 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
```

e. 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.

2. 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.

   a. 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. 1
5.1.168.192.in-addr.arpa. 604800 IN PTR api.ocp4.example.com. 2
```

1. Provides the record name for the Kubernetes internal API.
2. Provides 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.

7.9. GENERATING A KEY PAIR FOR CLUSTER NODE SSH ACCESS

During an OpenShift Container Platform installation, you can provide an SSH public key to the installation program. The key is passed to the Red Hat Enterprise Linux CoreOS (RHCOS) nodes through their Ignition config files and is used to authenticate SSH access to the nodes. The key is added to the **~/.ssh/authorized_keys** list for the **core** user on each node, which enables password-less authentication.

After the key is passed to the nodes, 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 such as AWS key pairs.

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> 1
   ```

   1 Specify 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 FIPS validated or Modules In Process cryptographic libraries on 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> 1
```

1. Specify 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.

7.10. VMWARE VSPHERE REGION AND ZONE ENABLEMENT

You can deploy an OpenShift Container Platform cluster to multiple vSphere datacenters that run in a single VMware vCenter. Each datacenter 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**
>
> VMware vSphere region and zone enablement 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 .

The default installation configuration deploys a cluster to a single vSphere datacenter. If you want to deploy a cluster to multiple vSphere datacenters, 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 datacenters 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 datacenter.

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

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

Table 7.12. Example of a configuration with multiple vSphere datacenters that run in a single VMware vCenter

Datacenter (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

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

2. Select your infrastructure provider.

3. Navigate to the page for your installation type, download the installation program that corresponds with your host operating system and architecture, and place the file in the directory where you will store the installation configuration files.

> **IMPORTANT**
>
> The installation program creates several files on the computer that you use to install your cluster. You must keep the installation program and the files that the installation program creates after you finish installing the cluster. Both files are required to delete the cluster.

> **IMPORTANT**
>
> 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.

4. 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
```

5. Download your installation pull secret from the 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.

7.12. MANUALLY CREATING THE INSTALLATION CONFIGURATION FILE

Installing the cluster requires that you manually create the installation configuration file.

Prerequisites

- You have an SSH public key on your local machine to provide to the installation program. The key will be used 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, 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.

2. Customize the sample **install-config.yaml** file template that is provided and save it 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 multiple clusters.

IMPORTANT

The **install-config.yaml** file is consumed during the next step of the installation process. You must back it up now.

7.12.1. 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.

```
apiVersion: v1
baseDomain: example.com 1
compute: 2
  name: worker
  replicas: 0 3
controlPlane: 4
  name: master
  replicas: 3 5
metadata:
  name: test 6
platform:
  vsphere:
    vcenter: your.vcenter.server 7
    username: username 8
    password: password 9
    datacenter: datacenter 10
    defaultDatastore: datastore 11
    folder: "/<datacenter_name>/vm/<folder_name>/<subfolder_name>" 12
    resourcePool: "/<datacenter_name>/host/<cluster_name>/Resources/<resource_pool_name>" 13
    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. Although both sections currently define a single machine pool, it is possible that future versions of OpenShift Container Platform will support defining multiple compute pools during installation. Only one control plane pool is used.

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 The fully-qualified hostname or IP address of the vCenter server.

IMPORTANT

The Cluster 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.

8 The name of the user for accessing the server.

9 The password associated with the vSphere user.

10 The vSphere datacenter.

11 The default vSphere datastore to use.

12 Optional parameter: For installer-provisioned infrastructure, the absolute path of an existing folder where the installation program creates the virtual machines, for example, **/<datacenter_name>/vm/<folder_name>/<subfolder_name>**. If you do not provide this value, the installation program creates a top-level folder in the datacenter 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.

13 Optional parameter: For installer-provisioned infrastructure, the absolute path of an existing folder where the installation program creates the virtual machines, for example, **/<datacenter_name>/vm/<folder_name>/<subfolder_name>**. If you do not provide this value, the installation program creates a top-level folder in the datacenter virtual machine folder that is named with the infrastructure ID. If you are providing the infrastructure for the cluster, omit this parameter.

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

> **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 Installing the system in FIPS mode. The use of FIPS validated or Modules In Process cryptographic libraries is only supported on OpenShift Container Platform deployments on the **x86_64**, **ppc64le**, and **s390x** architectures.

16 The pull secret that you obtained from OpenShift Cluster Manager Hybrid Cloud Console . 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).

7.12.2. Configuring the cluster-wide proxy during installation

Production environments can deny direct access to the internet and instead have an HTTP or HTTPS proxy available. You can configure a new OpenShift Container Platform cluster to use a proxy by configuring the proxy settings in the **install-config.yaml** file.

Prerequisites

- You have an existing **install-config.yaml** file.
- You 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 Platform (GCP), 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> 1
  httpsProxy: https://<username>:<pswd>@<ip>:<port> 2
  noProxy: example.com 3
additionalTrustBundle: | 4
    -----BEGIN CERTIFICATE-----
    <MY_TRUSTED_CA_CERT>
    -----END CERTIFICATE-----
additionalTrustBundlePolicy: <policy_to_add_additionalTrustBundle> 5
```

1. A proxy URL to use for creating HTTP connections outside the cluster. The URL scheme must be **http**.
2. A proxy URL to use for creating HTTPS connections outside the cluster.
3. 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.
4. If provided, the installation program generates a config map that is named **user-ca-bundle** in the **openshift-config** namespace that contains one or more additional CA certificates that are required for proxying HTTPS connections. The Cluster Network Operator then creates a **trusted-ca-bundle** config map that merges these contents with the Red Hat Enterprise Linux CoreOS (RHCOS) trust bundle, and this config map is referenced in the **trustedCA** field of the **Proxy** object. The **additionalTrustBundle** field is required unless the proxy's identity certificate is signed by an authority from the RHCOS trust bundle.
5. Optional: The policy to determine 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**.

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.

7.12.3. Configuring regions and zones for a VMware vCenter

You can modify the default installation configuration file to deploy an OpenShift Container Platform cluster to multiple vSphere datacenters that run in a single VMware vCenter.

IMPORTANT

VMware vSphere region and zone enablement 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.

IMPORTANT

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

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 datacenter objects for your VMware vCenter server. Consider specifying multiple failure domains if you need to provision virtual machine nodes in different datacenters, clusters, datastores, and other components. To enable regions and zones, you must define multiple failure domains for your OpenShift Container Platform cluster.

  NOTE

  You cannot change a failure domain after you installed an OpenShift Container Platform cluster on the VMware vSphere platform. You can add additional failure domains after cluster installation.

Procedure

1. Enter the following **govc** command-line tool commands to create the **openshift-region** and **openshift-zone** vCenter tag categories:

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. To create a region tag for each region vSphere datacenter where you want to deploy your cluster, enter the following command in your terminal:

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

3. To create a zone tag for each vSphere cluster where you want to deploy your cluster, enter the following command:

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

4. Attach region tags to each vCenter datacenter object by entering the following command:

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

5. Attach the zone tags to each vCenter datacenter object by entering the following command:

```
$ govc tags.attach -c <zone_tag_category> <zone_tag_1> /<datacenter_1>/host/vcs-mdcnc-workload-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 datacenters defined in a vSphere center

```
apiVersion: v1
baseDomain: example.com
featureSet: TechPreviewNoUpgrade 1
compute:
  name: worker
  replicas: 3
  vsphere:
    zones: 2
      - "<machine_pool_zone_1>"
      - "<machine_pool_zone_2>"
controlPlane:
  name: master
  replicas: 3
  vsphere:
    zones: 3
      - "<machine_pool_zone_1>"
      - "<machine_pool_zone_2>"
metadata:
```

```
  name: cluster
platform:
  vsphere:
    vcenter: <vcenter_server> 4
    username: <username> 5
    password: <password> 6
    datacenter: datacenter 7
    defaultDatastore: datastore 8
    folder: "/<datacenter_name>/vm/<folder_name>/<subfolder_name>" 9
    cluster: cluster 10
    resourcePool: "/<datacenter_name>/host/<cluster_name>/Resources/<resource_pool_name>" 11
    diskType: thin
    failureDomains: 12
    - name: <machine_pool_zone_1> 13
      region: <region_tag_1> 14
      zone: <zone_tag_1> 15
      topology: 16
        datacenter: <datacenter1> 17
        computeCluster: "/<datacenter1>/host/<cluster1>" 18
        resourcePool: "/<datacenter1>/host/<cluster1>/Resources/<resourcePool1>" 19
        networks: 20
        - <VM_Network1_name>
        datastore: "/<datacenter1>/datastore/<datastore1>" 21
    - name: <machine_pool_zone_2>
      region: <region_tag_2>
      zone: <zone_tag_2>
      topology:
        datacenter: <datacenter2>
        computeCluster: "/<datacenter2>/host/<cluster2>"
        networks:
        - <VM_Network2_name>
        datastore: "/<datacenter2>/datastore/<datastore2>"
        resourcePool: "/<datacenter2>/host/<cluster2>/Resources/<resourcePool2>"
        folder: "/<datacenter2>/vm/<folder2>"
# ...
```

1 You must define set the **TechPreviewNoUpgrade** as the value for this parameter, so that you can use the VMware vSphere region and zone enablement feature.

2 3 An optional parameter for specifying a vCenter cluster. You define a zone by using a tag from the **openshift-zone** tag category. If you do not define this parameter, nodes will be distributed among all defined failure-domains.

4 5 6 7 8 9 10 11 The default vCenter topology. The installation program uses this topology information to deploy the bootstrap node. Additionally, the topology defines the default datastore for vSphere persistent volumes.

12 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. If you do not define this parameter, the installation program uses the default vCenter topology.

13 Defines the name of the failure domain. Each failure domain is referenced in the **zones** parameter to scope a machine pool to the failure domain.

14 You define a region by using a tag from the **openshift-region** tag category. The tag must be attached to the vCenter datacenter.

15 You define a zone by using a tag from the **openshift-zone tag** category. The tag must be attached to the vCenter datacenter.

16 Specifies the vCenter resources associated with the failure domain.

17 An optional parameter for defining the vSphere datacenter that is associated with a failure domain. If you do not define this parameter, the installation program uses the default vCenter topology.

18 An optional parameter for stating the absolute file path for the compute cluster that is associated with the failure domain. If you do not define this parameter, the installation program uses the default vCenter topology.

19 An optional parameter for the installer-provisioned infrastructure. The parameter sets the absolute path of an existing resource pool where the installation program creates the virtual machines, for example, **/<datacenter_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_datacenter/host/example_cluster/Resources**.

20 An optional parameter that lists any network in the vCenter instance that contains the virtual IP addresses and DNS records that you configured. If you do not define this parameter, the installation program uses the default vCenter topology.

21 An optional parameter for specifying a datastore to use for provisioning volumes. If you do not define this parameter, the installation program uses the default vCenter topology.

7.13. 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.yml** 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 examples:

Specify a different VXLAN port for the OpenShift SDN network provider

```
apiVersion: operator.openshift.io/v1
kind: Network
metadata:
  name: cluster
spec:
  defaultNetwork:
    openshiftSDNConfig:
      vxlanPort: 4800
```

Enable IPsec for the OVN-Kubernetes network provider

```
apiVersion: operator.openshift.io/v1
kind: Network
metadata:
  name: cluster
spec:
  defaultNetwork:
    ovnKubernetesConfig:
      ipsecConfig: {}
```

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.

7.14. CLUSTER NETWORK OPERATOR CONFIGURATION

The configuration for the cluster network is specified as part of the Cluster Network Operator (CNO) configuration and stored in a custom resource (CR) object that is named **cluster**. The CR specifies the fields for the **Network** API in the **operator.openshift.io** API group.

The CNO configuration inherits the following fields during cluster installation from the **Network** API in the **Network.config.openshift.io** API group and these fields cannot be changed:

clusterNetwork
: IP address pools from which pod IP addresses are allocated.

serviceNetwork
: IP address pool for services.

defaultNetwork.type
: Cluster network plugin, such as OpenShift SDN or OVN-Kubernetes.

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**.

7.14.1. Cluster Network Operator configuration object

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

Table 7.13. 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**	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:
		`spec:`
		`  clusterNetwork:`
		`  - cidr: 10.128.0.0/19`
		`    hostPrefix: 23`
		`  - cidr: 10.128.32.0/19`
		`    hostPrefix: 23`
		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.

spec.serviceNetwork	**array**	A block of IP addresses for services. The OpenShift SDN and OVN-Kubernetes network plugins support only a single IP address block for the service network. For example:
		`spec:`
		`  serviceNetwork:`
		`  - 172.30.0.0/14`
		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.

Field	Type	Description
spec.defaultNetwork	**object**	Configures the network plugin for the cluster network.
spec.kubeProxyConfig	**object**	The fields for this object specify the kube-proxy configuration. If you are using the OVN-Kubernetes cluster network plugin, the kube-proxy configuration has no effect.

defaultNetwork object configuration

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

Table 7.14. **defaultNetwork** object

Field	Type	Description
type	**string**	Either **OpenShiftSDN** or **OVNKubernetes**. The Red Hat OpenShift Networking network plugin is selected during installation. This value cannot be changed after cluster installation. NOTE OpenShift Container Platform uses the OVN-Kubernetes network plugin by default.
openshiftSDNConfig	**object**	This object is only valid for the OpenShift SDN network plugin.
ovnKubernetesConfig	**object**	This object is only valid for the OVN-Kubernetes network plugin.

Configuration for the OpenShift SDN network plugin

The following table describes the configuration fields for the OpenShift SDN network plugin:

Table 7.15. **openshiftSDNConfig** object

Field	Type	Description
mode	**string**	Configures the network isolation mode for OpenShift SDN. The default value is **NetworkPolicy**. The values **Multitenant** and **Subnet** are available for backwards compatibility with OpenShift Container Platform 3.x but are not recommended. This value cannot be changed after cluster installation.

Field	Type	Description
mtu	**integer**	The maximum transmission unit (MTU) for the VXLAN 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. 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. If your cluster requires different MTU values for different nodes, you must set this value to **50** 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 **1450**. This value cannot be changed after cluster installation.
vxlanPort	**integer**	The port to use for all VXLAN packets. The default value is **4789**. This value cannot be changed after cluster installation. If you are running in a virtualized environment with existing nodes that are part of another VXLAN network, then you might be required to change this. For example, when running an OpenShift SDN overlay on top of VMware NSX-T, you must select an alternate port for the VXLAN, because both SDNs use the same default VXLAN port number. On Amazon Web Services (AWS), you can select an alternate port for the VXLAN between port **9000** and port **9999**.

Example OpenShift SDN configuration

```
defaultNetwork:
  type: OpenShiftSDN
  openshiftSDNConfig:
    mode: NetworkPolicy
    mtu: 1450
    vxlanPort: 4789
```

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

Table 7.16. ovnKubernetesConfig object

Field	Type	Description

Field	Type	Description
mtu	**integer**	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. 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. 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**.
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 an empty object to enable IPsec encryption.
policyAuditConfig	**object**	Specify a configuration object for customizing network policy audit logging. If unset, the defaults audit log settings are used.
gatewayConfig	**object**	Optional: Specify a configuration object for customizing how egress traffic is sent to the node gateway. NOTE 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.

Field	Type	Description
v4InternalSubnet	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**. This field cannot be changed after installation.	The default value is **100.64.0.0/16**.

Field	Type	Description
v6InternalSubnet	If your existing network infrastructure overlaps with the **fd98::/48** 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. This field cannot be changed after installation.	The default value is **fd98::/48**.

Table 7.17. **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
destination	string	One of the following additional audit log targets: **libc** The libc **syslog()** function of the journald process on the host. **udp:<host>:<port>** A syslog server. Replace **<host>:<port>** with the host and port of the syslog server. **unix:<file>** A Unix Domain Socket file specified by **<file>**. **null** Do not send the audit logs to any additional target.
syslogFacility	string	The syslog facility, such as **kern**, as defined by RFC5424. The default value is **local0**.

Table 7.18. **gatewayConfig** object

Field	Type	Description
routingViaHost	**boolean**	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**. 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.

Example OVN-Kubernetes configuration with IPSec enabled

```
defaultNetwork:
  type: OVNKubernetes
  ovnKubernetesConfig:
    mtu: 1400
    genevePort: 6081
    ipsecConfig: {}
```

kubeProxyConfig object configuration

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

Table 7.19. **kubeProxyConfig** object

<table>
<tr><th>Field</th><th>Type</th><th>Description</th></tr>
<tr><td>iptablesSyncPeriod</td><td>string</td><td>The refresh period for iptables rules. The default value is 30s. Valid suffixes include s, m, and h and are described in the Go time package documentation.

NOTE

Because of performance improvements introduced in OpenShift Container Platform 4.3 and greater, adjusting the iptablesSyncPeriod parameter is no longer necessary.</td></tr>
<tr><td>proxyArguments.iptables-min-sync-period</td><td>array</td><td>The minimum duration before refreshing iptables rules. This field ensures that the refresh does not happen too frequently. Valid suffixes include s, m, and h and are described in the Go time package. The default value is:<pre>kubeProxyConfig:
  proxyArguments:
    iptables-min-sync-period:
    - 0s</pre></td></tr>
</table>

7.15. 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. For a restricted network installation, these files are on your mirror host.

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
```

7.16. EXTRACTING THE INFRASTRUCTURE NAME

The Ignition config files contain a unique cluster identifier that you can use to uniquely identify your cluster in VMware Cloud on AWS. 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.

7.17. 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 datacenter to store your VMs.
   a. Click the **VMs and Templates** view.
   b. Right-click the name of your datacenter.

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.

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 datacenter.

   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.

7.18. 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 datacenter.
5. On the **Select storage** tab, select storage for your configuration and disk files.
6. On the **Select clone options**, select **Customize this virtual machine's hardware**.
7. On the **Customize hardware** tab, click **Advanced**.
   a. Click **Edit Configuration**, and on the **Configuration Parameters** window, click **Add Configuration Params**. Define the following parameter names and values:
      - **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.

7.19. 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.

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: 4.12.0
metadata:
  labels:
    machineconfiguration.openshift.io/role: worker
  name: 98-var-partition
storage:
  disks:
  - device: /dev/<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

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.

7.20. WAITING FOR THE BOOTSTRAP PROCESS TO COMPLETE

The OpenShift Container Platform bootstrap process begins after the cluster nodes first boot into the persistent RHCOS environment that has been installed to disk. The configuration information provided through the Ignition config files is used to initialize the bootstrap process and install OpenShift Container Platform on the machines. You must wait for the bootstrap process to complete.

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 \ 1
    --log-level=info 2
```

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

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

Example output

```
INFO Waiting up to 30m0s for the Kubernetes API at https://api.test.example.com:6443...
INFO API v1.25.0 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.

7.21. LOGGING IN TO THE CLUSTER BY USING THE CLI

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

Prerequisites

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

Procedure

1. Export the **kubeadmin** credentials:

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

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

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

```
$ oc whoami
```

Example output

Example output

```
system:admin
```

7.22. APPROVING THE CERTIFICATE SIGNING REQUESTS FOR YOUR MACHINES

When you add machines to a cluster, two pending certificate signing requests (CSRs) are generated for each machine that you added. You must confirm that these CSRs are approved or, if necessary, approve them yourself. The client requests must be approved 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  v1.25.0
   master-1  Ready     master  63m  v1.25.0
   master-2  Ready     master  64m  v1.25.0
   ```

   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> 1
     ```

     1 **<csr_name>** is 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> 1
     ```

     1 **<csr_name>** is 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  v1.25.0
   master-1  Ready     master  73m  v1.25.0
   master-2  Ready     master  74m  v1.25.0
   worker-0  Ready     worker  11m  v1.25.0
   worker-1  Ready     worker  11m  v1.25.0
   ```

   NOTE

   It can take a few minutes after approval of the server CSRs for the machines to transition to the **Ready** status.

Additional information

- For more information on CSRs, see Certificate Signing Requests.

7.23. INITIAL OPERATOR CONFIGURATION

After the control plane initializes, you must immediately configure some Operators so that they all become available.

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                        4.12.0    True        False         False      19m
baremetal                             4.12.0    True        False         False      37m
cloud-credential                      4.12.0    True        False         False      40m
cluster-autoscaler                    4.12.0    True        False         False      37m
config-operator                       4.12.0    True        False         False      38m
console                               4.12.0    True        False         False      26m
csi-snapshot-controller               4.12.0    True        False         False      37m
dns                                   4.12.0    True        False         False      37m
etcd                                  4.12.0    True        False         False      36m
image-registry                        4.12.0    True        False         False      31m
ingress                               4.12.0    True        False         False      30m
insights                              4.12.0    True        False         False      31m
kube-apiserver                        4.12.0    True        False         False      26m
kube-controller-manager               4.12.0    True        False         False      36m
kube-scheduler                        4.12.0    True        False         False      36m
kube-storage-version-migrator         4.12.0    True        False         False      37m
machine-api                           4.12.0    True        False         False      29m
machine-approver                      4.12.0    True        False         False      37m
machine-config                        4.12.0    True        False         False      36m
marketplace                           4.12.0    True        False         False      37m
monitoring                            4.12.0    True        False         False      29m
network                               4.12.0    True        False         False      38m
node-tuning                           4.12.0    True        False         False      37m
openshift-apiserver                   4.12.0    True        False         False      32m
openshift-controller-manager          4.12.0    True        False         False      30m
openshift-samples                     4.12.0    True        False         False      32m
operator-lifecycle-manager            4.12.0    True        False         False      37m
operator-lifecycle-manager-catalog    4.12.0    True        False         False      37m
operator-lifecycle-manager-packageserver  4.12.0    True        False         False      32m
service-ca                            4.12.0    True        False         False      38m
storage                               4.12.0    True        False         False      37m
```

2. Configure the Operators that are not available.

7.23.1. 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.

7.23.2. 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.

Instructions are shown for configuring a persistent volume, which is required for production clusters. Where applicable, instructions are shown for configuring an empty directory as the storage location, which is available for only non-production clusters.

Additional instructions are provided for allowing the image registry to use block storage types by using the **Recreate** rollout strategy during upgrades.

7.23.2.1. 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

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 PV for the block storage device, and create a 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 1
  namespace: openshift-image-registry 2
spec:
  accessModes:
  - ReadWriteOnce 3
  resources:
    requests:
      storage: 100Gi 4
```

1. A unique name that represents the **PersistentVolumeClaim** object.

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

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

4 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: 1
```

1 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.

7.24. COMPLETING INSTALLATION ON USER-PROVISIONED INFRASTRUCTURE

After you complete the Operator configuration, you can finish installing the cluster on 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                             4.12.0    True        False         False      19m
baremetal                                  4.12.0    True        False         False      37m
cloud-credential                           4.12.0    True        False         False      40m
cluster-autoscaler                         4.12.0    True        False         False      37m
config-operator                            4.12.0    True        False         False      38m
console                                    4.12.0    True        False         False      26m
```

```
csi-snapshot-controller                    4.12.0   True      False        False     37m
dns                                        4.12.0   True      False        False     37m
etcd                                       4.12.0   True      False        False     36m
image-registry                             4.12.0   True      False        False     31m
ingress                                    4.12.0   True      False        False     30m
insights                                   4.12.0   True      False        False     31m
kube-apiserver                             4.12.0   True      False        False     26m
kube-controller-manager                    4.12.0   True      False        False     36m
kube-scheduler                             4.12.0   True      False        False     36m
kube-storage-version-migrator              4.12.0   True      False        False     37m
machine-api                                4.12.0   True      False        False     29m
machine-approver                           4.12.0   True      False        False     37m
machine-config                             4.12.0   True      False        False     36m
marketplace                                4.12.0   True      False        False     37m
monitoring                                 4.12.0   True      False        False     29m
network                                    4.12.0   True      False        False     38m
node-tuning                                4.12.0   True      False        False     37m
openshift-apiserver                        4.12.0   True      False        False     32m
openshift-controller-manager               4.12.0   True      False        False     30m
openshift-samples                          4.12.0   True      False        False     32m
operator-lifecycle-manager                 4.12.0   True      False        False     37m
operator-lifecycle-manager-catalog         4.12.0   True      False        False     37m
operator-lifecycle-manager-packageserver   4.12.0   True      False        False     32m
service-ca                                 4.12.0   True      False        False     38m
storage                                    4.12.0   True      False        False     37m
```

Alternatively, the following command notifies you when all of the clusters are available. It also retrieves and displays credentials:

```
$ ./openshift-install --dir <installation_directory> wait-for install-complete 1
```

1 For **<installation_directory>**, specify 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> 1
   ```

   1 Specify the pod name and namespace, as shown in the output of the previous 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 *Post-installation 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.

7.25. BACKING UP VMWARE VSPHERE VOLUMES

OpenShift Container Platform provisions new volumes as independent persistent disks to freely attach and detach the volume on any node in the cluster. As a consequence, it is not possible to back up volumes that use snapshots, or to restore volumes from snapshots. See Snapshot Limitations for more information.

Procedure

To create a backup of persistent volumes:

1. Stop the application that is using the persistent volume.
2. Clone the persistent volume.
3. Restart the application.
4. Create a backup of the cloned volume.
5. Delete the cloned volume.

7.26. TELEMETRY ACCESS FOR OPENSHIFT CONTAINER PLATFORM

In OpenShift Container Platform 4.12, the Telemetry service, which runs by default to provide metrics about cluster health and the success of updates, requires internet access. If your cluster is connected to the internet, Telemetry runs automatically, and your cluster is registered to OpenShift Cluster Manager Hybrid Cloud Console.

After you confirm that your OpenShift Cluster Manager Hybrid Cloud Console 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.

Additional resources

- See About remote health monitoring for more information about the Telemetry service

7.27. NEXT STEPS

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

CHAPTER 8. INSTALLING A CLUSTER ON VMC IN A RESTRICTED NETWORK WITH USER-PROVISIONED INFRASTRUCTURE

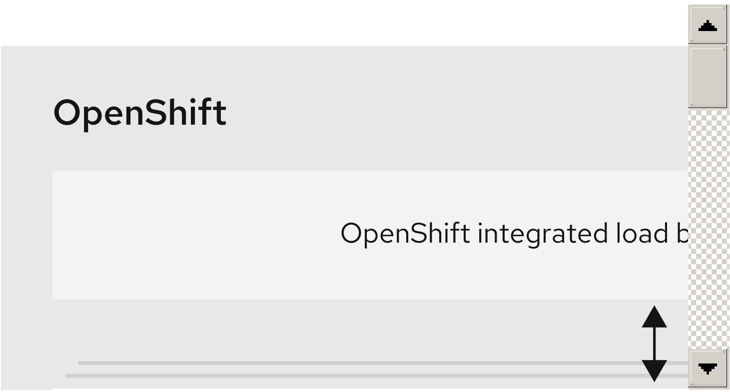
In OpenShift Container Platform version 4.12, you can install a cluster on VMware vSphere infrastructure that you provision in a restricted network by deploying it to VMware Cloud (VMC) on AWS.

Once you configure your VMC environment for OpenShift Container Platform deployment, you use the OpenShift Container Platform installation program from the bastion management host, co-located in the VMC environment. The installation program and control plane automates the process of deploying and managing the resources needed for the OpenShift Container Platform cluster.

> **NOTE**
>
> OpenShift Container Platform supports deploying a cluster to a single VMware vCenter only. Deploying a cluster with machines/machine sets on multiple vCenters is not supported.

8.1. SETTING UP VMC FOR VSPHERE

You can install OpenShift Container Platform on VMware Cloud (VMC) on AWS hosted vSphere clusters to enable applications to be deployed and managed both on-premise and off-premise, across the hybrid cloud.



You must configure several options in your VMC environment prior to installing OpenShift Container Platform on VMware vSphere. Ensure your VMC environment has the following prerequisites:

- Create a non-exclusive, DHCP-enabled, NSX-T network segment and subnet. Other virtual machines (VMs) can be hosted on the subnet, but at least eight IP addresses must be available for the OpenShift Container Platform deployment.
- Configure the following firewall rules:
  - An ANY:ANY firewall rule between the installation host and the software-defined data center (SDDC) management network on port 443. This allows you to upload the Red Hat Enterprise Linux CoreOS (RHCOS) OVA during deployment.
  - An HTTPS firewall rule between the OpenShift Container Platform compute network and vCenter. This connection allows OpenShift Container Platform to communicate with vCenter for provisioning and managing nodes, persistent volume claims (PVCs), and other resources.

- You must have the following information to deploy OpenShift Container Platform:
    - The OpenShift Container Platform cluster name, such as **vmc-prod-1**.
    - The base DNS name, such as **companyname.com**.
    - If not using the default, the pod network CIDR and services network CIDR must be identified, which are set by default to **10.128.0.0/14** and **172.30.0.0/16**, respectively. These CIDRs are used for pod-to-pod and pod-to-service communication and are not accessible externally; however, they must not overlap with existing subnets in your organization.
    - The following vCenter information:
        - vCenter hostname, username, and password
        - Datacenter name, such as **SDDC-Datacenter**
        - Cluster name, such as **Cluster-1**
        - Network name
        - Datastore name, such as **WorkloadDatastore**

        NOTE

        It is recommended to move your vSphere cluster to the VMC **Compute-ResourcePool** resource pool after your cluster installation is finished.
- A Linux-based host deployed to VMC as a bastion.
    - The bastion host can be Red Hat Enterprise Linux (RHEL) or any another Linux-based host; it must have internet connectivity and the ability to upload an OVA to the ESXi hosts.
    - Download and install the OpenShift CLI tools to the bastion host.
        - The **openshift-install** installation program
        - The OpenShift CLI (**oc**) tool

NOTE

You cannot use the VMware NSX Container Plugin for Kubernetes (NCP), and NSX is not used as the OpenShift SDN. The version of NSX currently available with VMC is incompatible with the version of NCP certified with OpenShift Container Platform.

However, the NSX DHCP service is used for virtual machine IP management with the full-stack automated OpenShift Container Platform deployment and with nodes provisioned, either manually or automatically, by the Machine API integration with vSphere. Additionally, NSX firewall rules are created to enable access with the OpenShift Container Platform cluster and between the bastion host and the VMC vSphere hosts.

8.1.1. VMC Sizer tool

VMware Cloud on AWS is built on top of AWS bare metal infrastructure; this is the same bare metal infrastructure which runs AWS native services. When a VMware cloud on AWS software-defined data center (SDDC) is deployed, you consume these physical server nodes and run the VMware ESXi hypervisor in a single tenant fashion. This means the physical infrastructure is not accessible to anyone else using VMC. It is important to consider how many physical hosts you will need to host your virtual infrastructure.

To determine this, VMware provides the VMC on AWS Sizer. With this tool, you can define the resources you intend to host on VMC:

- Types of workloads
- Total number of virtual machines
- Specification information such as:
  - Storage requirements
  - vCPUs
  - vRAM
  - Overcommit ratios

With these details, the sizer tool can generate a report, based on VMware best practices, and recommend your cluster configuration and the number of hosts you will need.

8.2. VSPHERE PREREQUISITES

- 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 obtain 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 block registry storage. For more information on persistent storage, see Understanding persistent storage.
- 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.

8.3. ABOUT INSTALLATIONS IN RESTRICTED NETWORKS

In OpenShift Container Platform 4.12, 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.

8.3.1. 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.

8.4. INTERNET ACCESS FOR OPENSHIFT CONTAINER PLATFORM

In OpenShift Container Platform 4.12, you require access to the internet to obtain the images that are necessary to install your cluster.

You must have internet access to:

- Access OpenShift Cluster Manager Hybrid Cloud Console 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.

8.5. 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 7.0 Update 2 or later
- Version 8.0 Update 1 or later

You can host the VMware vSphere infrastructure on-premise or on a VMware Cloud Verified provider that meets the requirements outlined in the following table:

Table 8.1. Version requirements for vSphere virtual environments

Virtual environment product	Required version
VMware virtual hardware	15 or later
vSphere ESXi hosts	7.0 Update 2 or later; 8.0 Update 1 or later
vCenter host	7.0 Update 2 or later; 8.0 Update 1 or later

IMPORTANT

Installing a cluster on VMware vSphere versions 7.0 and 7.0 Update 1 is deprecated. These versions are still fully supported, but all vSphere 6.x versions are no longer supported. Version 4.12 of OpenShift Container Platform requires VMware virtual hardware version 15 or later. To update the hardware version for your vSphere virtual machines, see the "Updating hardware on nodes running in vSphere" article in the *Updating clusters* section.

Table 8.2. Minimum supported vSphere version for VMware components

Component	Minimum supported versions	Description
Hypervisor	vSphere 7.0 Update 2 (or later) or vSphere 8.0 Update 1 (or later) with virtual hardware version 15	This 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.
Storage with in-tree drivers	vSphere 7.0 Update 2 or later; 8.0 Update 1 or later	This plugin creates vSphere storage by using the in-tree storage drivers for vSphere included in OpenShift Container Platform.

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.

8.6. VMWARE VSPHERE CSI DRIVER OPERATOR REQUIREMENTS

To install the vSphere CSI Driver Operator, the following requirements must be met:

- VMware vSphere version: 7.0 Update 2 or later; 8.0 Update 1 or later
- vCenter version: 7.0 Update 2 or later; 8.0 Update 1 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.

Additional resources

- To remove a third-party 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.

8.7. 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.

8.7.1. vCenter requirements

Before you install an OpenShift Container Platform cluster on your vCenter that uses infrastructure that you provided, you must prepare your environment.

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 8.1. 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 will be created in the cluster root	**Host.Config.Storage** **Resource.AssignVMToPool** **VApp.AssignResourcePool** **VApp.Import** **VirtualMachine.Config.AddNewDisk**
vSphere vCenter Resource Pool	If an existing resource pool is provided	**Host.Config.Storage** **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.Add**

vSphere object for role	When required	Required privileges in vSphere API
		RemoveDevice **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** **VirtualMachine.Config.ResetGuestInfo** **VirtualMachine.Config.Resource** **VirtualMachine.Config.Settings** **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 Datacenter	If the installation program creates the virtual machine folder. For UPI, **VirtualMachine.Inventory.Create** and **VirtualMachine.Inventory.D**	**InventoryService.Tagging.ObjectAttachable** **Resource.AssignVMToPool** **VApp.Import** **VirtualMachine.Config.AddExistingDisk** **VirtualMachine.Config.AddNewDisk**

vSphere object for role	When required	Required privileges in vSphere API
	elete privileges are optional if your cluster does not use the Machine API.	**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** **VirtualMachine.Config.ResetGuestInfo** **VirtualMachine.Config.Resource** **VirtualMachine.Config.Settings** **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.DeployTemplate** **VirtualMachine.Provisioning.MarkAsTemplate** **Folder.Create** **Folder.Delete**

Example 8.2. Roles and privileges required for installation in vCenter graphical user interface (GUI)

vSphere object for role	When required	Required privileges in vCenter GUI
vSphere 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" "Profile-driven storage"."Profile-driven storage update" "Profile-driven storage"."Profile-driven storage view"**
vSphere vCenter Cluster	If VMs will be created 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 an existing resource pool is provided	**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 object for role	When required	Required privileges in vCenter GUI
vSphere Datastore	Always	**Datastore."Allocate space" Datastore."Browse datastore" Datastore."Low level file operations" "vSphere Tagging"."Assign or Unassign vSphere Tag on Object"**
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"**

vSphere object for role	When required	Required privileges in vCenter GUI
		"Virtual machine"."Change Configuration".Rename" **"Virtual machine"."Change Configuration"."Reset guest information"** **"Virtual machine"."Change Configuration"."Change resource"** **"Virtual machine"."Change Configuration"."Change Settings"** **"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 Datacenter	If the installation program creates the virtual machine folder. For UPI, **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"**

vSphere object for role	When required	Required privileges in vCenter GUI
		"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 Settings" "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"

vSphere object for role	When required	Required privileges in vCenter GUI
		"Virtual machine"."Edit Inventory"."Create from existing" **"Virtual machine"."Edit Inventory"."Remove"** **"Virtual machine".Provisioning."Clone virtual machine"** **"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 propogate the permissions to child objects. These settings vary depending on whether or not you install the cluster into an existing folder.

Example 8.3. Required permissions and propagation settings

vSphere object	When required	Propagate to children	Permissions required
vSphere vCenter	Always	False	Listed required privileges
vSphere vCenter Datacenter	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 object	When required	Propagate to children	Permissions required
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.

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.

- OpenShift Container Platform generally supports compute-only vMotion, 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.

- Using Storage vMotion can cause issues and is not supported. If you are using 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 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.

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 is destroyed during the cluster installation process. A minimum of 800 GB of storage is required to use a standard cluster.

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

Cluster limits

Available resources vary between clusters. The number of possible clusters within a vCenter is limited 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.

Networking requirements

You must use the Dynamic Host Configuration Protocol (DHCP) for the network and ensure that the DHCP server is configured to provide persistent IP addresses to the cluster machines. In the DHCP lease, you must configure the DHCP to use the default gateway. All nodes must be in the same VLAN. You cannot scale the cluster using a second VLAN as a Day 2 operation. Additionally, you must create the following networking resources before you install the OpenShift Container Platform cluster:

> **NOTE**
>
> It is recommended that each OpenShift Container Platform node in the cluster must have access to a Network Time Protocol (NTP) server that is discoverable via DHCP. Installation is possible without an NTP server. However, asynchronous server clocks will cause errors, which NTP server prevents.

Required IP Addresses

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 8.3. Required DNS records

Component	Record	Description
API VIP	**api.<cluster_name>.<base_domain>.**	This DNS A/AAAA or CNAME 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	***.apps.<cluster_name>.<base_domain>.**	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

8.7.2. Required machines for cluster installation

The smallest OpenShift Container Platform clusters require the following hosts:

Table 8.4. 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.
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) 8 and inherits all of its hardware certifications and requirements. See Red Hat Enterprise Linux technology capabilities and limits .

8.7.3. Minimum resource requirements for cluster installation

Each cluster machine must meet the following minimum requirements:

Table 8.5. Minimum resource requirements

Machine	Operating System	vCPU [1]	Virtual RAM	Storage	Input/Output Per Second (IOPS)[2]
Bootstrap	RHCOS	4	16 GB	100 GB	300
Control plane	RHCOS	4	16 GB	100 GB	300
Compute	RHCOS, RHEL 8.6 and later [3]	2	8 GB	100 GB	300

Machine	Operating System	vCPU [1]	Virtual RAM	Storage	Input/Output Per Second (IOPS)[2]

1. One vCPU is equivalent to one physical core when simultaneous multithreading (SMT), or hyperthreading, is not enabled. When enabled, use the following formula to calculate the corresponding ratio: (threads per core × cores) × sockets = vCPUs.

2. 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.

3. 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.

If an instance type for your platform meets the minimum requirements for cluster machines, it is supported to use in OpenShift Container Platform.

8.7.4. Certificate signing requests management

Because your cluster has limited access to automatic machine management when you use infrastructure that you provision, you must provide a mechanism for approving cluster certificate signing requests (CSRs) after installation. 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.

8.7.5. Networking requirements for user-provisioned infrastructure

All the Red Hat Enterprise Linux CoreOS (RHCOS) machines require networking to be configured in **initramfs** during boot to fetch their Ignition config files.

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.

It is recommended to use 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.

NOTE

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.

8.7.5.1. 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.

8.7.5.2. 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 8.6. 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

Protocol	Port	Description
	10256	openshift-sdn
UDP	**4789**	VXLAN
	6081	Geneve
	9000-9999	Host level services, including the node exporter on ports **9100-9101**.
	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	IPsec Encapsulating Security Payload (ESP)

Table 8.7. Ports used for all-machine to control plane communications

Protocol	Port	Description
TCP	**6443**	Kubernetes API

Table 8.8. Ports used for control plane machine to control plane machine communications

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

Ethernet adaptor hardware address requirements

When provisioning VMs for the cluster, the ethernet interfaces configured for each VM must use a MAC address from the VMware Organizationally Unique Identifier (OUI) allocation ranges:

- **00:05:69:00:00:00** to **00:05:69:FF:FF:FF**
- **00:0c:29:00:00:00** to **00:0c:29:FF:FF:FF**
- **00:1c:14:00:00:00** to **00:1c:14:FF:FF:FF**
- **00:50:56:00:00:00** to **00:50:56:3F:FF:FF**

If a MAC address outside the VMware OUI is used, the cluster installation will not succeed.

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.

8.7.6. User-provisioned DNS requirements

In OpenShift Container Platform deployments, DNS name resolution is required for the following 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 8.9. Required DNS records

Component	Record	Description
Kubernetes API	**api.<cluster_name>.<base_domain>.**	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.

Component	Record	Description
	api-int.<cluster_name>.<base_domain>.	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.
Routes	***.apps.<cluster_name>.<base_domain>.**	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, **console-openshift-console.apps.<cluster_name>.<base_domain>** is used as a wildcard route to the OpenShift Container Platform console.
Bootstrap machine	**bootstrap.<cluster_name>.<base_domain>.**	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	**<control_plane><n>.<cluster_name>.<base_domain>.**	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	**<compute><n>.<cluster_name>.<base_domain>.**	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.

8.7.6.1. Example DNS configuration for user-provisioned clusters

This section provides A and PTR record configuration samples that meet the DNS requirements for deploying OpenShift Container Platform on user-provisioned infrastructure. The samples 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**.

Example DNS A record configuration for a user-provisioned cluster

The following example is a BIND zone file that shows sample A records for name resolution in a user-provisioned cluster.

Example 8.4. Sample DNS zone database

```
$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 1
api-int.ocp4.example.com. IN A 192.168.1.5 2
;
*.apps.ocp4.example.com. IN A 192.168.1.5 3
;
bootstrap.ocp4.example.com. IN A 192.168.1.96 4
;
control-plane0.ocp4.example.com. IN A 192.168.1.97 5
control-plane1.ocp4.example.com. IN A 192.168.1.98 6
control-plane2.ocp4.example.com. IN A 192.168.1.99 7
;
compute0.ocp4.example.com. IN A 192.168.1.11 8
compute1.ocp4.example.com. IN A 192.168.1.7 9
;
;EOF
```

1. Provides name resolution for the Kubernetes API. The record refers to the IP address of the API load balancer.

2. 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.

3. 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

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.

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.

4 Provides name resolution for the bootstrap machine.

5 6 7 Provides name resolution for the control plane machines.

8 9 Provides name resolution for the compute machines.

Example DNS PTR record configuration for a user-provisioned cluster

The following example BIND zone file shows sample PTR records for reverse name resolution in a user-provisioned cluster.

Example 8.5. Sample DNS zone database for reverse records

```
$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. 1
5.1.168.192.in-addr.arpa. IN PTR api-int.ocp4.example.com. 2
;
96.1.168.192.in-addr.arpa. IN PTR bootstrap.ocp4.example.com. 3
;
97.1.168.192.in-addr.arpa. IN PTR control-plane0.ocp4.example.com. 4
98.1.168.192.in-addr.arpa. IN PTR control-plane1.ocp4.example.com. 5
99.1.168.192.in-addr.arpa. IN PTR control-plane2.ocp4.example.com. 6
;
11.1.168.192.in-addr.arpa. IN PTR compute0.ocp4.example.com. 7
7.1.168.192.in-addr.arpa. IN PTR compute1.ocp4.example.com. 8
;
;EOF
```

1 Provides reverse DNS resolution for the Kubernetes API. The PTR record refers to the record name of the API load balancer.

2 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.

3 Provides reverse DNS resolution for the bootstrap machine.

4 5 6 Provides reverse DNS resolution for the control plane machines.

7 8 Provides reverse DNS resolution for the compute machines.

NOTE

A PTR record is not required for the OpenShift Container Platform application wildcard.

8.7.7. 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:

1. **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 load balancers:

   Table 8.10. API load balancer

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

Port	Back-end machines (pool members)	Internal	External	Description
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.

2. **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 8.11. 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.

8.7.7.1. Example load balancer configuration for user-provisioned clusters

This section provides an example API and application ingress load balancer configuration that meets 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.

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**.

Example 8.6. 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 1
  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 2
```

```
  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 3
  bind *:22623
  mode tcp
  server bootstrap bootstrap.ocp4.example.com:22623 check inter 1s backup 4
  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 5
  bind *:443
  mode tcp
  balance source
  server worker0 worker0.ocp4.example.com:443 check inter 1s
  server worker1 worker1.ocp4.example.com:443 check inter 1s
listen ingress-router-80 6
  bind *:80
  mode tcp
  balance source
  server worker0 worker0.ocp4.example.com:80 check inter 1s
  server worker1 worker1.ocp4.example.com:80 check inter 1s
```

1. Port **6443** handles the Kubernetes API traffic and points to the control plane machines.

2. 4. 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.

3. Port **22623** handles the machine config server traffic and points to the control plane machines.

5. 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.

6. 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.

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 -nltupe** on the HAProxy node.

8.8. PREPARING THE USER-PROVISIONED INFRASTRUCTURE

Before you install OpenShift Container Platform on user-provisioned infrastructure, you must prepare the underlying infrastructure.

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

1. If you are using DHCP to provide the IP networking configuration to your cluster nodes, configure your DHCP service.
    a. 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.

8.9. VALIDATING DNS RESOLUTION FOR USER-PROVISIONED INFRASTRUCTURE

You can validate your DNS configuration before installing OpenShift Container Platform on user-provisioned infrastructure.

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> 1
   ```

   1 Replace **<nameserver_ip>** with the IP address of the nameserver, **<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
   ```

   c. 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
```

d. 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
```

e. 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.

2. 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.

a. 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. (1)
5.1.168.192.in-addr.arpa. 604800 IN PTR api.ocp4.example.com. (2)
```

(1) Provides the record name for the Kubernetes internal API.

(2) Provides 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.

8.10. GENERATING A KEY PAIR FOR CLUSTER NODE SSH ACCESS

During an OpenShift Container Platform installation, you can provide an SSH public key to the installation program. The key is passed to the Red Hat Enterprise Linux CoreOS (RHCOS) nodes through their Ignition config files and is used to authenticate SSH access to the nodes. The key is added to the **~/.ssh/authorized_keys** list for the **core** user on each node, which enables password-less authentication.

After the key is passed to the nodes, 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 such as AWS key pairs.

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> 1
   ```

   1 Specify 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 FIPS validated or Modules In Process cryptographic libraries on 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> (1)
```

(1) Specify 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.

8.11. VMWARE VSPHERE REGION AND ZONE ENABLEMENT

You can deploy an OpenShift Container Platform cluster to multiple vSphere datacenters that run in a single VMware vCenter. Each datacenter 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

VMware vSphere region and zone enablement 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 .

The default installation configuration deploys a cluster to a single vSphere datacenter. If you want to deploy a cluster to multiple vSphere datacenters, 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 datacenters 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 datacenter.

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

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

Table 8.12. Example of a configuration with multiple vSphere datacenters that run in a single VMware vCenter

Datacenter (region)	Cluster (zone)	Tags
us-east	us-east-1	us-east-1a

Datacenter (region)	Cluster (zone)	Tags
		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

8.12. MANUALLY CREATING THE INSTALLATION CONFIGURATION FILE

Installing the cluster requires that you manually create the installation configuration file.

Prerequisites

- You have an SSH public key on your local machine to provide to the installation program. The key will be used 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, 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.

2. Customize the sample **install-config.yaml** file template that is provided and save it 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.

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 next step of the installation process. You must back it up now.

8.12.1. 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.

```
apiVersion: v1
baseDomain: example.com 1
compute: 2
  name: worker
  replicas: 0 3
controlPlane: 4
  name: master
  replicas: 3 5
metadata:
  name: test 6
platform:
  vsphere:
    vcenter: your.vcenter.server 7
    username: username 8
    password: password 9
    datacenter: datacenter 10
    defaultDatastore: datastore 11
    folder: "/<datacenter_name>/vm/<folder_name>/<subfolder_name>" 12
    resourcePool: "/<datacenter_name>/host/<cluster_name>/Resources/<resource_pool_name>" 13
    diskType: thin 14
fips: false 15
pullSecret: '{"auths":{"<local_registry>": {"auth": "<credentials>","email": "you@example.com"}}}' 16
sshKey: 'ssh-ed25519 AAAA...' 17
additionalTrustBundle: | 18
  -----BEGIN CERTIFICATE-----
  ZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZ
```

```
  -----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. Although both sections currently define a single machine pool, it is possible that future versions of OpenShift Container Platform will support defining multiple compute pools during installation. Only one control plane pool is used.

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. The fully-qualified hostname or IP address of the vCenter server.

IMPORTANT

The Cluster 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.

8. The name of the user for accessing the server.

9. The password associated with the vSphere user.

10. The vSphere datacenter.

11. The default vSphere datastore to use.

12. Optional parameter: For installer-provisioned infrastructure, the absolute path of an existing folder where the installation program creates the virtual machines, for example, **/<datacenter_name>/vm/<folder_name>/<subfolder_name>**. If you do not provide this value, the installation program creates a top-level folder in the datacenter 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.

13. Optional parameter: For installer-provisioned infrastructure, the absolute path of an existing folder

where the installation program creates the virtual machines, for example, **/<datacenter_name>/vm/<folder_name>/<subfolder_name>**. If you do not provide this value, the installation program creates a top-level folder in the datacenter virtual machine folder that is named with the infrastructure ID. If you are providing the infrastructure for the cluster, omit this parameter.

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 Installing the system in FIPS mode. The use of FIPS validated or Modules In Process cryptographic libraries is only supported on OpenShift Container Platform deployments on 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.

8.12.2. Configuring the cluster-wide proxy during installation

Production environments can deny direct access to the internet and instead have an HTTP or HTTPS proxy available. You can configure a new OpenShift Container Platform cluster to use a proxy by configuring the proxy settings in the **install-config.yaml** file.

Prerequisites

- You have an existing **install-config.yaml** file.
- You 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 Platform (GCP), 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> 1
  httpsProxy: https://<username>:<pswd>@<ip>:<port> 2
  noProxy: example.com 3
additionalTrustBundle: | 4
    -----BEGIN CERTIFICATE-----
    <MY_TRUSTED_CA_CERT>
    -----END CERTIFICATE-----
additionalTrustBundlePolicy: <policy_to_add_additionalTrustBundle> 5
```

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

2. A proxy URL to use for creating HTTPS connections outside the cluster.

3. 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.

4. If provided, the installation program generates a config map that is named **user-ca-bundle** in the **openshift-config** namespace that contains one or more additional CA certificates that are required for proxying HTTPS connections. The Cluster Network Operator then creates a **trusted-ca-bundle** config map that merges these contents with the Red Hat Enterprise Linux CoreOS (RHCOS) trust bundle, and this config map is referenced in the **trustedCA** field of the **Proxy** object. The **additionalTrustBundle** field is required unless the proxy's identity certificate is signed by an authority from the RHCOS trust bundle.

5. Optional: The policy to determine 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**.

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.

8.12.3. Configuring regions and zones for a VMware vCenter

You can modify the default installation configuration file to deploy an OpenShift Container Platform cluster to multiple vSphere datacenters that run in a single VMware vCenter.

IMPORTANT

VMware vSphere region and zone enablement 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 .

IMPORTANT

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

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 datacenter objects for your VMware vCenter server. Consider specifying multiple failure domains if you need to provision virtual machine nodes in different datacenters, clusters, datastores, and other components. To enable regions and zones, you must define multiple failure domains for your OpenShift Container Platform cluster.

NOTE

You cannot change a failure domain after you installed an OpenShift Container Platform cluster on the VMware vSphere platform. You can add additional failure domains after cluster installation.

Procedure

1. Enter the following **govc** command-line tool commands to create the **openshift-region** and **openshift-zone** vCenter tag categories:

    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. To create a region tag for each region vSphere datacenter where you want to deploy your cluster, enter the following command in your terminal:

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

3. To create a zone tag for each vSphere cluster where you want to deploy your cluster, enter the following command:

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

4. Attach region tags to each vCenter datacenter object by entering the following command:

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

5. Attach the zone tags to each vCenter datacenter object by entering the following command:

    ```
    $ govc tags.attach -c <zone_tag_category> <zone_tag_1> /<datacenter_1>/host/vcs-mdcnc-workload-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 datacenters defined in a vSphere center

```yaml
apiVersion: v1
baseDomain: example.com
featureSet: TechPreviewNoUpgrade # 1
compute:
  name: worker
  replicas: 3
  vsphere:
    zones: # 2
      - "<machine_pool_zone_1>"
      - "<machine_pool_zone_2>"
controlPlane:
  name: master
  replicas: 3
  vsphere:
    zones: # 3
      - "<machine_pool_zone_1>"
      - "<machine_pool_zone_2>"
metadata:
  name: cluster
platform:
  vsphere:
    vcenter: <vcenter_server> # 4
    username: <username> # 5
    password: <password> # 6
    datacenter: datacenter # 7
    defaultDatastore: datastore # 8
    folder: "/<datacenter_name>/vm/<folder_name>/<subfolder_name>" # 9
    cluster: cluster # 10
    resourcePool: "/<datacenter_name>/host/<cluster_name>/Resources/<resource_pool_name>" # 11
    diskType: thin
    failureDomains: # 12
    - name: <machine_pool_zone_1> # 13
      region: <region_tag_1> # 14
      zone: <zone_tag_1> # 15
      topology: # 16
        datacenter: <datacenter1> # 17
        computeCluster: "/<datacenter1>/host/<cluster1>" # 18
        resourcePool: "/<datacenter1>/host/<cluster1>/Resources/<resourcePool1>" # 19
        networks: # 20
        - <VM_Network1_name>
        datastore: "/<datacenter1>/datastore/<datastore1>" # 21
    - name: <machine_pool_zone_2>
      region: <region_tag_2>
      zone: <zone_tag_2>
      topology:
        datacenter: <datacenter2>
        computeCluster: "/<datacenter2>/host/<cluster2>"
        networks:
        - <VM_Network2_name>
        datastore: "/<datacenter2>/datastore/<datastore2>"
```

```
      resourcePool: "/<datacenter2>/host/<cluster2>/Resources/<resourcePool2>"
      folder: "/<datacenter2>/vm/<folder2>"
# ...
```

1 You must define set the **TechPreviewNoUpgrade** as the value for this parameter, so that you can use the VMware vSphere region and zone enablement feature.

2 3 An optional parameter for specifying a vCenter cluster. You define a zone by using a tag from the **openshift-zone** tag category. If you do not define this parameter, nodes will be distributed among all defined failure-domains.

4 5 6 7 8 9 10 11 The default vCenter topology. The installation program uses this topology information to deploy the bootstrap node. Additionally, the topology defines the default datastore for vSphere persistent volumes.

12 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. If you do not define this parameter, the installation program uses the default vCenter topology.

13 Defines the name of the failure domain. Each failure domain is referenced in the **zones** parameter to scope a machine pool to the failure domain.

14 You define a region by using a tag from the **openshift-region** tag category. The tag must be attached to the vCenter datacenter.

15 You define a zone by using a tag from the **openshift-zone tag** category. The tag must be attached to the vCenter datacenter.

16 Specifies the vCenter resources associated with the failure domain.

17 An optional parameter for defining the vSphere datacenter that is associated with a failure domain. If you do not define this parameter, the installation program uses the default vCenter topology.

18 An optional parameter for stating the absolute file path for the compute cluster that is associated with the failure domain. If you do not define this parameter, the installation program uses the default vCenter topology.

19 An optional parameter for the installer-provisioned infrastructure. The parameter sets the absolute path of an existing resource pool where the installation program creates the virtual machines, for example, **/<datacenter_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_datacenter/host/example_cluster/Resources**.

20 An optional parameter that lists any network in the vCenter instance that contains the virtual IP addresses and DNS records that you configured. If you do not define this parameter, the installation program uses the default vCenter topology.

21 An optional parameter for specifying a datastore to use for provisioning volumes. If you do not define this parameter, the installation program uses the default vCenter topology.

8.13. CREATING THE KUBERNETES MANIFEST AND IGNITION CONFIG FILES

Because you must modify some cluster definition files and manually start the cluster machines, you must generate the Kubernetes manifest and Ignition config files that the cluster needs to configure the machines.

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> 1
```

1 For **<installation_directory>**, specify the installation directory that contains the **install-config.yaml** file you created.

2. Remove the Kubernetes manifest files that define the control plane machines and compute machine sets:

```
$ 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 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.yml** 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.yml** 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> 1
```

1 For **<installation_directory>**, specify 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
```

8.14. EXTRACTING THE INFRASTRUCTURE NAME

The Ignition config files contain a unique cluster identifier that you can use to uniquely identify your cluster in VMware Cloud on AWS. 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.

8.15. 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 datacenter to store your VMs.
   a. Click the **VMs and Templates** view.
   b. Right-click the name of your datacenter.
   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.

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 datacenter.

   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.

8.16. 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 datacenter.
5. On the **Select storage** tab, select storage for your configuration and disk files.
6. On the **Select clone options**, select **Customize this virtual machine's hardware**.
7. On the **Customize hardware** tab, click **Advanced**.
   a. Click **Edit Configuration**, and on the **Configuration Parameters** window, click **Add Configuration Params**. Define the following parameter names and values:
      - **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.

8.17. 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.

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: 4.12.0
metadata:
  labels:
    machineconfiguration.openshift.io/role: worker
  name: 98-var-partition
storage:
  disks:
  - device: /dev/<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

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.

8.18. WAITING FOR THE BOOTSTRAP PROCESS TO COMPLETE

The OpenShift Container Platform bootstrap process begins after the cluster nodes first boot into the persistent RHCOS environment that has been installed to disk. The configuration information provided through the Ignition config files is used to initialize the bootstrap process and install OpenShift Container Platform on the machines. You must wait for the bootstrap process to complete.

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 \ 1
    --log-level=info 2
```

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

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

Example output

```
INFO Waiting up to 30m0s for the Kubernetes API at https://api.test.example.com:6443...
INFO API v1.25.0 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.

8.19. LOGGING IN TO THE CLUSTER BY USING THE CLI

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

Prerequisites

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

Procedure

1. Export the **kubeadmin** credentials:

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

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

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

```
$ oc whoami
```

Example output

Example output

```
system:admin
```

8.20. APPROVING THE CERTIFICATE SIGNING REQUESTS FOR YOUR MACHINES

When you add machines to a cluster, two pending certificate signing requests (CSRs) are generated for each machine that you added. You must confirm that these CSRs are approved or, if necessary, approve them yourself. The client requests must be approved 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  v1.25.0
master-1  Ready     master  63m  v1.25.0
master-2  Ready     master  64m  v1.25.0
```

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> 1
     ```

     1 **<csr_name>** is 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> 1
     ```

     1 **<csr_name>** is 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  v1.25.0
   master-1  Ready     master  73m  v1.25.0
   master-2  Ready     master  74m  v1.25.0
   worker-0  Ready     worker  11m  v1.25.0
   worker-1  Ready     worker  11m  v1.25.0
   ```

   NOTE

   It can take a few minutes after approval of the server CSRs for the machines to transition to the **Ready** status.

Additional information

- For more information on CSRs, see Certificate Signing Requests.

8.21. INITIAL OPERATOR CONFIGURATION

After the control plane initializes, you must immediately configure some Operators so that they all become available.

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                          4.12.0    True        False         False      19m
baremetal                               4.12.0    True        False         False      37m
cloud-credential                        4.12.0    True        False         False      40m
cluster-autoscaler                      4.12.0    True        False         False      37m
config-operator                         4.12.0    True        False         False      38m
console                                 4.12.0    True        False         False      26m
csi-snapshot-controller                 4.12.0    True        False         False      37m
dns                                     4.12.0    True        False         False      37m
etcd                                    4.12.0    True        False         False      36m
image-registry                          4.12.0    True        False         False      31m
ingress                                 4.12.0    True        False         False      30m
insights                                4.12.0    True        False         False      31m
kube-apiserver                          4.12.0    True        False         False      26m
kube-controller-manager                 4.12.0    True        False         False      36m
kube-scheduler                          4.12.0    True        False         False      36m
kube-storage-version-migrator           4.12.0    True        False         False      37m
machine-api                             4.12.0    True        False         False      29m
machine-approver                        4.12.0    True        False         False      37m
machine-config                          4.12.0    True        False         False      36m
marketplace                             4.12.0    True        False         False      37m
monitoring                              4.12.0    True        False         False      29m
network                                 4.12.0    True        False         False      38m
node-tuning                             4.12.0    True        False         False      37m
openshift-apiserver                     4.12.0    True        False         False      32m
openshift-controller-manager            4.12.0    True        False         False      30m
openshift-samples                       4.12.0    True        False         False      32m
operator-lifecycle-manager              4.12.0    True        False         False      37m
operator-lifecycle-manager-catalog      4.12.0    True        False         False      37m
operator-lifecycle-manager-packageserver 4.12.0   True        False         False      32m
service-ca                              4.12.0    True        False         False      38m
storage                                 4.12.0    True        False         False      37m
```

2. Configure the Operators that are not available.

8.21.1. 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.

8.21.2. 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.

Instructions are shown for configuring a persistent volume, which is required for production clusters. Where applicable, instructions are shown for configuring an empty directory as the storage location, which is available for only non-production clusters.

Additional instructions are provided for allowing the image registry to use block storage types by using the **Recreate** rollout strategy during upgrades.

8.21.2.1. 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. To configure your registry to use storage, change the **spec.storage.pvc** 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:

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

   Example output

```
No resourses 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:

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

   Example output

```
storage:
  pvc:
    claim: 1
```

   1 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:

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

Example output

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

8.21.2.2. 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 for only 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.

8.21.2.3. 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 PV for the block storage device, and create a 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 1
      namespace: openshift-image-registry 2
    spec:
      accessModes:
      - ReadWriteOnce 3
      resources:
        requests:
          storage: 100Gi 4
    ```

    1 A unique name that represents the **PersistentVolumeClaim** object.

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

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

    4 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: 1
    ```

    1 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 registry storage for VMware vSphere.

8.22. COMPLETING INSTALLATION ON USER-PROVISIONED INFRASTRUCTURE

After you complete the Operator configuration, you can finish installing the cluster on 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                             4.12.0    True        False         False      19m
baremetal                                  4.12.0    True        False         False      37m
cloud-credential                           4.12.0    True        False         False      40m
cluster-autoscaler                         4.12.0    True        False         False      37m
config-operator                            4.12.0    True        False         False      38m
console                                    4.12.0    True        False         False      26m
csi-snapshot-controller                    4.12.0    True        False         False      37m
dns                                        4.12.0    True        False         False      37m
etcd                                       4.12.0    True        False         False      36m
image-registry                             4.12.0    True        False         False      31m
ingress                                    4.12.0    True        False         False      30m
insights                                   4.12.0    True        False         False      31m
kube-apiserver                             4.12.0    True        False         False      26m
kube-controller-manager                    4.12.0    True        False         False      36m
kube-scheduler                             4.12.0    True        False         False      36m
kube-storage-version-migrator              4.12.0    True        False         False      37m
machine-api                                4.12.0    True        False         False      29m
machine-approver                           4.12.0    True        False         False      37m
machine-config                             4.12.0    True        False         False      36m
marketplace                                4.12.0    True        False         False      37m
monitoring                                 4.12.0    True        False         False      29m
network                                    4.12.0    True        False         False      38m
node-tuning                                4.12.0    True        False         False      37m
openshift-apiserver                        4.12.0    True        False         False      32m
openshift-controller-manager               4.12.0    True        False         False      30m
openshift-samples                          4.12.0    True        False         False      32m
operator-lifecycle-manager                 4.12.0    True        False         False      37m
operator-lifecycle-manager-catalog         4.12.0    True        False         False      37m
operator-lifecycle-manager-packageserver   4.12.0    True        False         False      32m
service-ca                                 4.12.0    True        False         False      38m
storage                                    4.12.0    True        False         False      37m
```

Alternatively, the following command notifies you when all of the clusters are available. It also retrieves and displays credentials:

```
$ ./openshift-install --dir <installation_directory> wait-for install-complete (1)
```

(1) For **<installation_directory>**, specify 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> (1)
```

(1) Specify the pod name and namespace, as shown in the output of the previous 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 *Post-installation 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.

8.23. BACKING UP VMWARE VSPHERE VOLUMES

OpenShift Container Platform provisions new volumes as independent persistent disks to freely attach and detach the volume on any node in the cluster. As a consequence, it is not possible to back up volumes that use snapshots, or to restore volumes from snapshots. See Snapshot Limitations for more information.

Procedure

To create a backup of persistent volumes:

1. Stop the application that is using the persistent volume.
2. Clone the persistent volume.
3. Restart the application.
4. Create a backup of the cloned volume.
5. Delete the cloned volume.

8.24. TELEMETRY ACCESS FOR OPENSHIFT CONTAINER PLATFORM

In OpenShift Container Platform 4.12, the Telemetry service, which runs by default to provide metrics about cluster health and the success of updates, requires internet access. If your cluster is connected to the internet, Telemetry runs automatically, and your cluster is registered to OpenShift Cluster Manager Hybrid Cloud Console.

After you confirm that your OpenShift Cluster Manager Hybrid Cloud Console 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.

Additional resources

- See About remote health monitoring for more information about the Telemetry service

8.25. NEXT STEPS

- Customize your cluster.
- Configure image streams for the Cluster Samples Operator and the **must-gather** tool.
- Learn how to use Operator Lifecycle Manager (OLM) on restricted networks .
- 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 opt out of remote health reporting .
- Optional: View the events from the vSphere Problem Detector Operator to determine if the cluster has permission or storage configuration issues.

CHAPTER 9. UNINSTALLING A CLUSTER ON VMC

You can remove a cluster installed on VMware vSphere infrastructure that you deployed to VMware Cloud (VMC) on AWS by using installer-provisioned infrastructure.

9.1. REMOVING A CLUSTER THAT USES INSTALLER-PROVISIONED INFRASTRUCTURE

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

> **NOTE**
>
> After uninstallation, check your cloud provider for any resources not removed properly, especially with user-provisioned infrastructure clusters. There might be resources that the installation program did not create or that the installation program is unable to access.

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. On the computer that you used to install the cluster, go to the directory that contains the installation program, and run the following command:

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

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

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

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

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