OpenShift Container Platform 4.6

在 IBM Z 和 LinuxONE 上安装

安装 OpenShift Container Platform IBM Z 集群

Last Updated: 2023-02-27

OpenShift Container Platform 4.6 在 IBM Z 和 LinuxONE 上安装

安装 OpenShift Container Platform IBM Z 集群

Enter your first name here. Enter your surname here.
Enter your organisation's name here. Enter your organisational division here.
Enter your email address here.

法律通告

摘要

本文档提供在 IBM Z 上安装 OpenShift Container Platform 集群的说明。

目录

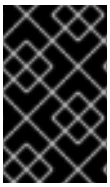
第 1 章 在 IBM Z 上安装

1.1. 在 IBM Z 和 LINUXONE 上安装集群

在 OpenShift Container Platform 版本 4.6 中，您可以在您置备的 IBM Z 或 LinuxONE 系统上安装集群。

注意

虽然本文档只提到了 IBM Z，但它所提供的所有信息同样也适用于 LinuxONE。



重要

非裸机平台还有其他注意事项。在尝试在此类环境中安装 OpenShift Container Platform 集群前，请参阅有关在未经测试的平台上部署 OpenShift Container Platform 的指南 中的信息。

1.1.1. 先决条件

- 在开始安装进程前，您必须清理安装目录。这可保证在安装过程中创建和更新所需的安装文件。
- 为集群置备使用 NFS 的持久性存储。若要部署私有镜像 registry，您的存储必须提供 **ReadWriteMany** 访问模式。
- 查看有关 OpenShift Container Platform 安装和更新 流程的详细信息。
- 如果使用防火墙，则必须将其配置为允许集群需要访问的站点。

注意

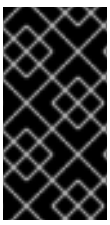
如果您要配置代理，请务必也要查看此站点列表。

1.1.2. OpenShift Container Platform 的互联网访问

在 OpenShift Container Platform 4.6 中，您需要访问互联网来安装集群。

您必须具有以下互联网访问权限：

- 访问 OpenShift Cluster Manager 以下载安装程序并执行订阅管理。如果集群可以访问互联网，并且没有禁用 Telemetry，该服务会自动授权您的集群。
- 访问 Quay.io，以获取安装集群所需的软件包。
- 获取执行集群更新所需的软件包。



重要

如果您的集群无法直接访问互联网，则可以在置备的某些类基础架构上执行受限网络安装。在此过程中，您要下载所需的内容，并使用它在镜像 registry（mirror registry） 中填充安装集群并生成安装程序所需的软件包。对于某些安装类型，集群要安装到的环境不需要访问互联网。在更新集群之前，要更新 registry 镜像系统中的内容。

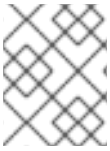
1.1.3. 具有用户置备基础架构的集群的机器要求

对于含有用户置备的基础架构的集群，您必须部署所有所需的机器。

1.1.3.1. 所需的机器

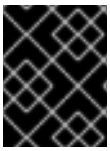
最小的 OpenShift Container Platform 集群需要下列主机：

- 一个临时 bootstrap 机器
- 三台 control plane 或 master 机器
- 至少两台计算机器，也称为 worker 机器。



注意

集群要求 bootstrap 机器在三台 control plane 机器上部署 OpenShift Container Platform 集群。您可在安装集群后删除 bootstrap 机器。



重要

要提高集群的高可用性，请在最少两个物理集群中的不同的 z/VM 实例中分布运行 control plane 机器。

bootstrap 和 control plane 机器必须使用 Red Hat Enterprise Linux CoreOS (RHCOS) 作为操作系统。但是，计算机器可以在 Red Hat Enterprise Linux CoreOS(RHCOS)或 Red Hat Enterprise Linux(RHEL)7.9 间进行选择。

请注意，RHCOS 基于 Red Hat Enterprise Linux（RHEL） 8，并继承其所有硬件认证和要求。请查看Red Hat Enterprise Linux 技术功能及限制 。

1.1.3.2. 网络连接要求

所有 Red Hat Enterprise Linux CoreOS (RHCOS) 机器在启动过程中需要 **initramfs** 中的网络从 Machine Config Server 获取 Ignition 配置文件。机器被配置为使用静态 IP 地址。不需要 DHCP 服务器。另外，集群中的每个 OpenShift Container Platform 节点都必须有权访问网络时间协议（NTP）服务器。

1.1.3.3. IBM Z 网络连接要求

要在 z/VM 中安装 IBM Z，您需要使用第 2 层模式的单一 z/VM 虚拟 NIC。您还需要：

- 直接连接的 OSA 或 RoCE 网络适配器
- z/VM vSwitch 设置。对于首选的设置，请使用 OSA 链接聚合。

1.1.3.4. 最低资源要求

每台集群机器都必须满足以下最低要求：

机器	操作系统	vCPU [1]	虚拟内存	存储	IOPS
bootstrap	RHCOS	4	16 GB	100 GB	N/A
Control plane	RHCOS	4	16 GB	100 GB	N/A

机器	操作系统	vCPU [1]	虚拟内存	存储	IOPS
Compute	RHCOS	2	8 GB	100 GB	N/A

1. 当未启用并发多线程 (SMT) 或超线程时，一个 vCPU 相当于一个物理内核。启用后，使用以下公式来计算对应的比例：（每个内核数的线程）× sockets = vCPU。

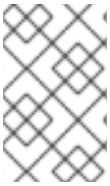
1.1.3.5. 最低 IBM Z 系统环境

您可以在以下 IBM 硬件上安装 OpenShift Container Platform 版本 4.6：

- IBM z15（所有型号）、IBM z14（所有型号）、IBM z13 和 IBM z13s
- 任何版本的 LinuxONE

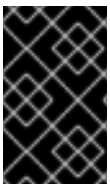
硬件要求

- 相当于为每个集群启用了 SMT2 的 IFL。
- 至少有一个网络连接连接到 **LoadBalancer** 服务，并为集群外的流量提供服务。



注意

您可以使用专用或共享 IFL 来分配充足的计算资源。资源共享是 IBM Z 的主要优势之一。但是，您必须在每个虚拟机监控程序层上正确调整容量，并确保每个 OpenShift Container Platform 集群有足够的资源。



重要

由于集群的整体性能可能会受到影响，因此用于设置 OpenShift Container Platform 集群的 LPAR 必须提供足够的计算能力。在这种情况下，虚拟机监控程序级别的 LPAR 权重管理、授权和 CPU 共享扮演重要角色。

操作系统要求

- 一个 z/VM 7.1 或更高版本的实例

在您的 z/VM 实例中设置：

- 3 个客户虚拟机作为 OpenShift Container Platform control plane 的机器
- 2 个客户虚拟机作为 OpenShift Container Platform 的计算机器
- 1 个客户虚拟机作为临时 OpenShift Container Platform bootstrap 机器

IBM Z 网络连接要求
要在 z/VM 中安装 IBM Z，您需要使用第 2 层模式的单一 z/VM 虚拟 NIC。您还需要：

- 直接连接的 OSA 或 RoCE 网络适配器
- z/VM vSwitch 设置。对于首选的设置，请使用 OSA 链接聚合。

z/VM 客户虚拟机的磁盘存储

- 附加了 FICON 的磁盘存储。可以是 z/VM Minidisks、fullpack Minidisks 或专用 DASD，它们都必须被格式化为 CDL，这是默认的 CDL。要达到 Red Hat Enterprise Linux CoreOS (RHCOS) 安装所需的最小 DASD 大小，您需要扩展地址卷 (EAV)。如果可用，使用 HyperPAV 来确保最佳性能。
- FCP 连接的磁盘存储

存储/主内存

- OpenShift Container Platform control plane 需要 16 GB
- OpenShift Container Platform 计算机器需要 8 GB
- 临时 OpenShift Container Platform bootstrap 机器需要 16 GB

1.1.3.6. 首选 IBM Z 系统环境

硬件要求

- 3 个与 6 个 IFL 等效的 LPARS，每个集群都启用了 SMT2。
- 用于连接 **LoadBalancer** 服务的两个网络连接，并为集群外的流量提供服务。
- HiperSockets，可直接作为设备附加到节点，或者与一个 z/VM VSWITCH 桥接以对 z/VM 客户机进行透明。要将 HiperSockets 直接连接到节点，您必须通过 RHEL 8 虚拟机设置到外部网络的网关来桥接到 HiperSockets 网络。

操作系统要求

- 2 个或 3 个 z/VM 7.1 或更高版本的实例以实现高可用性

在您的 z/VM 实例中设置：

- 3 个用于 OpenShift Container Platform control plane 机器的虚拟机，每个 z/VM 实例一个。
- 至少 6 个用于 OpenShift Container Platform 计算机器的虚拟机，分布在 z/VM 实例中。
- 1 个客户虚拟机作为临时 OpenShift Container Platform bootstrap 机器。
- 要确保在过量使用的环境中整合组件可用，请使用 CP 命令 **SET SHARE** 来增加 control plane 的优先级。对基础架构节点执行相同操作（如果存在）。请参阅 IBM 文档中的 SET SHARE。

IBM Z 网络连接要求
要在 z/VM 中安装 IBM Z，您需要使用第 2 层模式的单一 z/VM 虚拟 NIC。您还需要：

- 直接连接的 OSA 或 RoCE 网络适配器
- z/VM vSwitch 设置。对于首选的设置，请使用 OSA 链接聚合。

z/VM 客户虚拟机的磁盘存储

- 附加了 FICON 的磁盘存储。可以是 z/VM Minidisks、fullpack Minidisks 或专用 DASD，它们都必须被格式化为 CDL，这是默认的 CDL。要达到 Red Hat Enterprise Linux CoreOS (RHCOS) 安装所需的最小 DASD 大小，您需要扩展地址卷 (EAV)。如果可用，请使用 HyperPAV 和 High Performance FICON（zHPF）来确保最佳性能。
- FCP 连接的磁盘存储

存储/主内存

- OpenShift Container Platform control plane 需要 16 GB
- OpenShift Container Platform 计算机器需要 8 GB
- 临时 OpenShift Container Platform bootstrap 机器需要 16 GB

1.1.3.7. 证书签名请求管理

在使用您置备的基础架构时，集群只能有限地访问自动机器管理，因此您必须提供一种在安装后批准集群证书签名请求 (CSR) 的机制。**kube-controller-manager** 只能批准 kubelet 客户端 CSR。 **machine-approver** 无法保证使用 kubelet 凭证请求的提供证书的有效性，因为它不能确认是正确的机器发出了该请求。您必须决定并实施一种方法，以验证 kubelet 提供证书请求的有效性并进行批准。

其他资源

- 请参阅 IBM 文档中的使用 z/VM 虚拟交换机桥接 HiperSockets LAN。
- 请参阅在 z/VM 的 Linux 客户端中扩展 HyperPAV 别名设备 以获得性能优化。
- 有关 LPAR 权重管理和权利，请参阅 LPAR 性能中的主题。

1.1.4. 创建用户置备的基础架构

在部署采用用户置备的基础架构的 OpenShift Container Platform 集群前，您必须创建底层基础架构。

先决条件

- 在为集群创建支持基础架构之前，请参阅OpenShift Container Platform 4.x Tested Integrations页。

流程

1. 设置静态 IP 地址
2. 设置一个 FTP 服务器。
3. 提供所需的负载均衡器。
4. 配置机器的端口。
5. 配置 DNS。
6. 确保网络可以正常工作。

1.1.4.1. 用户置备的基础架构对网络的要求

所有 Red Hat Enterprise Linux CoreOS（RHCOS）机器在启动过程中需要 **initramfs** 中的网络从机器配置服务器获取 Ignition 配置。

在初次启动过程中，机器需要 FTP 服务器来建立网络连接，以下载其 Ignition 配置文件。

确保机器具有持久的 IP 地址和主机名。

Kubernetes API 服务器必须能够解析集群机器的节点名称。如果 API 服务器和 worker 节点位于不同的区域中，您可以配置默认 DNS 搜索区域，以便 API 服务器能够解析节点名称。另一种支持的方法是始终在节点对象和所有 DNS 请求中使用完全限定域名来指代主机。

您必须配置机器间的网络连接，以便集群组件进行通信。每台机器都必须能够解析集群中所有其他机器的主机名。

表 1.1. 所有机器到所有机器

协议	端口	描述
ICMP	N/A	网络可访问性测试
TCP	**1936**	指标
	9000-**9999**	主机级别的服务，包括端口 **9100**-**9101** 上的节点导出器和端口 **9099** 上的 Cluster Version Operator。
	10250-**10259**	Kubernetes 保留的默认端口
	10256	openshift-sdn
UDP	**4789**	VXLAN 和 Geneve
	6081	VXLAN 和 Geneve
	9000-**9999**	主机级别的服务，包括端口 **9100**-**9101** 上的节点导出器。
TCP/UDP	**30000**-**32767**	Kubernetes 节点端口

表 1.2. 要通过控制平面的所有机器

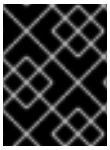
协议	端口	描述
TCP	**6443**	Kubernetes API

表 1.3. control plane 机器到 control plane 机器

协议	端口	描述
TCP	**2379**-**2380**	etcd 服务器和对等端口

网络拓扑要求

您为集群置备的基础架构必须满足下列网络拓扑要求。



重要

OpenShift Container Platform 要求所有节点都能访问互联网，以便为平台容器提取镜像并向红帽提供遥测数据。

负载均衡器

在安装 OpenShift Container Platform 前，您必须置备两个满足以下要求的负载均衡器：

1. **API 负载均衡器**：提供一个通用端点，供用户（包括人和机器）与平台交互和配置。配置以下条件：

   - 只适用于第 4 层负载均衡。这可被称为 Raw TCP、SSL Passthrough 或者 SSL 桥接模式。如果使用 SSL Bridge 模式，必须为 API 路由启用 Server Name Indication（SNI）。
   - 无状态负载平衡算法。这些选项根据负载均衡器的实现而有所不同。

   重要

   不要为 API 负载均衡器配置会话持久性。

   在负载均衡器的前端和后台配置以下端口：

   表 1.4. API 负载均衡器

端口	后端机器（池成员）	内部	外部	描述
6443	Bootstrap 和 control plane.bootstrap 机器初始化集群 control plane 后，您要从负载均衡器中删除 bootstrap 机器。您必须为 API 服务器健康检查探测配置 **/readyz** 端点。	X	X	Kubernetes API 服务器
22623	Bootstrap 和 control plane.bootstrap 机器初始化集群 control plane 后，您要从负载均衡器中删除 bootstrap 机器。	X		机器配置服务器

   注意

   负载均衡器必须配置为，从 API 服务器关闭 **/readyz** 端点到从池中删除 API 服务器实例时最多需要 30 秒。在 **/readyz** 返回错误或处于健康状态后的时间范围内，端点必须被删除或添加。每 5 秒或 10 秒探测一次，有两个成功请求处于健康状态，三个成为不健康的请求经过测试。

2. **应用程序入口负载均衡器**:提供来自集群外部的应用程序流量流量的 Ingress 点。配置以下条件：

   - 只适用于第 4 层负载均衡。这可被称为 Raw TCP、SSL Passthrough 或者 SSL 桥接模式。如果使用 SSL Bridge 模式，您必须为 Ingress 路由启用Server Name Indication（SNI）。
   - 建议根据可用选项以及平台上托管的应用程序类型，使用基于连接的或者基于会话的持久性。

   在负载均衡器的前端和后台配置以下端口：

   表 1.5. 应用程序入口负载均衡器

端口	后端机器（池成员）	内部	外部	描述
443	默认运行入口路由器 Pod、计算或 worker 的机器。	X	X	HTTPS 流量
80	默认运行入口路由器 Pod、计算或 worker 的机器。	X	X	HTTP 流量

提示

如果负载均衡器可以看到客户端的真实 IP 地址，启用基于 IP 的会话持久性可提高使用端到端 TLS 加密的应用程序的性能。

注意

OpenShift Container Platform 集群需要正确配置入口路由器。control plane 初始化后，您必须配置入口路由器。

NTP 配置

OpenShift Container Platform 集群默认配置为使用公共网络时间协议（NTP）服务器。如果要使用本地企业 NTP 服务器，或者集群部署在断开连接的网络中，您可以将集群配置为使用特定的时间服务器。如需更多信息，请参阅*配置 chrony 时间服务*的文档。

其他资源

- 配置 chrony 时间服务

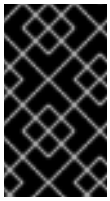
1.1.4.2. 用户置备 DNS 要求

DNS 用于名称解析和反向名称解析。DNS A/AAAA 或 CNAME 记录用于名称解析，PTR 记录用于反向解析名称。反向记录很重要，因为 Red Hat Enterprise Linux CoreOS（RHCOS）使用反向记录为所有节点设置主机名。另外，反向记录用于生成 OpenShift Container Platform 需要操作的证书签名请求（CSR）。

采用用户置备的基础架构的 OpenShift Container Platform 集群需要以下 DNS 记录。在每一记录中，**<cluster_name>** 是集群名称，**<base_domain>** 则是您在 **install-config.yaml** 文件中指定的集群基域。完整的 DNS 记录采用如下格式: **<component>.<cluster_name>.<base_domain>.**。

表 1.6. 所需的 DNS 记录

组件	记录	描述
Kubernetes API	**api.<cluster_name>.<base_domain>.**	添加 DNS A/AAAA 或 CNAME 记录，以及 DNS PTR 记录，以识别 control plane 机器的负载均衡器。这些记录必须由集群外的客户端以及集群中的所有节点解析。

组件	记录	描述
	api-int.<cluster_name>. <base_domain>.	添加 DNS A/AAAA 或 CNAME 记录，以及 DNS PTR 记录，以识别 control plane 机器的负载均衡器。这些记录必须可以从集群中的所有节点解析。  **重要** API 服务器必须能够根据在 Kubernetes 中记录的主机名解析 worker 节点。如果 API 服务器无法解析节点名称，则代理的 API 调用会失败，且您无法从 pod 检索日志。
Routes	***.apps.<cluster_name>. <base_domain>.**	添加通配符 DNS A/AAAA 或 CNAME 记录，指向以运行入口路由器 Pod 的机器（默认为 worker 节点）为目标的负载均衡器。这些记录必须由集群外的客户端以及集群中的所有节点解析。
bootstrap	**bootstrap.<cluster_name>. <base_domain>.**	添加 DNS A/AAAA 或 CNAME 记录，以及 DNS PTR 记录来识别 bootstrap 机器。这些记录必须由集群中的节点解析。
Master 主机	**<master><n>. <cluster_name>. <base_domain>.**	DNS A/AAAA 或 CNAME 记录，以识别 control plane 节点（也称为 master 节点）的每台机器。这些记录必须由集群中的节点解析。
Worker 主机	**<worker><n>. <cluster_name>. <base_domain>.**	添加 DNS A/AAAA 或 CNAME 记录，以识别 worker 节点的每台机器。这些记录必须由集群中的节点解析。

提示

您可以使用 **nslookup <hostname>** 命令来验证名称解析。您可以使用 **dig -x <ip_address>** 命令来验证 PTR 记录的反向名称解析。

下面的 BIND 区文件的例子展示了关于名字解析的 A 记录的例子。这个示例的目的是显示所需的记录。这个示例不是为选择一个名称解析服务提供建议。

例 1.1. DNS 区数据库示例

```
$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 IN A 192.168.1.5
```

```
smtp IN A 192.168.1.5
;
helper IN A 192.168.1.5
helper.ocp4 IN A 192.168.1.5
;
; The api identifies the IP of your load balancer.
api.ocp4  IN A 192.168.1.5
api-int.ocp4  IN A 192.168.1.5
;
; The wildcard also identifies the load balancer.
*.apps.ocp4  IN A 192.168.1.5
;
; Create an entry for the bootstrap host.
bootstrap.ocp4 IN A 192.168.1.96
;
; Create entries for the master hosts.
master0.ocp4  IN A 192.168.1.97
master1.ocp4  IN A 192.168.1.98
master2.ocp4  IN A 192.168.1.99
;
; Create entries for the worker hosts.
worker0.ocp4  IN A 192.168.1.11
worker1.ocp4  IN A 192.168.1.7
;
;EOF
```

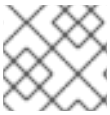
下面的 BIND 区文件示例显示了反向名字解析的 PTR 记录示例。

例 1.2. 反向记录的 DNS 区数据库示例

```
$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.
;
; The syntax is "last octet" and the host must have an FQDN
; with a trailing dot.
97 IN PTR master0.ocp4.example.com.
98 IN PTR master1.ocp4.example.com.
99 IN PTR master2.ocp4.example.com.
;
96 IN PTR bootstrap.ocp4.example.com.
;
5 IN PTR api.ocp4.example.com.
5 IN PTR api-int.ocp4.example.com.
;
11 IN PTR worker0.ocp4.example.com.
7 IN PTR worker1.ocp4.example.com.
;
;EOF
```

1.1.5. 生成 SSH 私钥并将其添加到代理中

如果要在集群上执行安装调试或灾难恢复，则必须为 **ssh-agent** 和安装程序提供 SSH 密钥。您可以使用此密钥访问公共集群中的 bootstrap 机器来排除安装问题。



注意

在生产环境中，您需要进行灾难恢复和调试。

重要

不要在生产环境中跳过这个过程，因为生产环境需要灾难恢复和调试。

您可以使用此密钥以 **core** 用户身份通过 SSH 连接到 master 节点。在部署集群时，此密钥会添加到 **core** 用户的 **~/.ssh/authorized_keys** 列表中。

流程

1. 如果还没有为计算机上免密码身份验证而配置的 SSH 密钥，请创建一个。例如，在使用 Linux 操作系统的计算机上运行以下命令：

```
$ ssh-keygen -t ed25519 -N '' \
  -f <path>/<file_name> 1
```

1 指定新 SSH 密钥的路径和文件名，如 **~/.ssh/id_rsa**。如果您已有密钥对，请确保您的公钥位于 **~/.ssh** 目录中。

运行此命令会在指定的位置生成不需要密码的 SSH 密钥。

注意

如果您计划在 **x86_64** 架构中安装使用 FIPS 验证的/Modules in Process 加密库的 OpenShift Container Platform 集群，不要创建使用 **ed25519** 算法的密钥。反之，创建一个使用 **rsa** 或 **ecdsa** 算法的密钥。

2. 作为后台任务启动 **ssh-agent** 进程：

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

输出示例

```
Agent pid 31874
```

注意

如果您的集群采用 FIPS 模式，则只使用 FIPS 兼容算法来生成 SSH 密钥。密钥必须是 RSA 或 ECDSA。

3. 将 SSH 私钥添加到 **ssh-agent**：

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

输出示例

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

1 指定 SSH 私钥的路径和文件名，如 **~/.ssh/id_rsa**

后续步骤

- 在安装 OpenShift Container Platform 时，为安装程序提供 SSH 公钥。

1.1.6. 获取安装程序

在安装 OpenShift Container Platform 之前，将安装文件下载到您置备的机器上。

先决条件

- 一个运行 Linux 的机器，如 Red Hat Enterprise Linux 8，本地磁盘空间为 500MB。

流程

1. 访问 OpenShift Cluster Manager 站点的 Infrastructure Provider 页面。如果您有红帽帐号，请使用自己的凭证登录。如果没有，请创建一个帐户。
2. 选择您的基础架构供应商。
3. 进入适用于您的安装类型的页面，下载您的操作系统的安装程序，并将文件放在要保存安装配置文件的目录中。。

重要

安装程序会在用来安装集群的计算机上创建若干文件。在完成集群安装后，您必须保留安装程序和安装程序所创建的文件。这两个文件都需要删除集群。

重要

删除安装程序创建的文件不会删除您的集群，即使集群在安装过程中失败也是如此。要删除集群，为特定云供应商完成 OpenShift Container Platform 卸载流程。

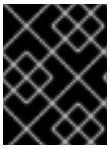
4. 提取安装程序。例如，在使用 Linux 操作系统的计算机上运行以下命令：

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

5. 从 Red Hat OpenShift Cluster Manager 下载安装 pull secret 。通过此 pull secret，您可以进行所含授权机构提供的服务的身份验证，这些服务包括为 OpenShift Container Platform 组件提供容器镜像的 Quay.io。

1.1.7. 通过下载二进制文件安装 OpenShift CLI

您需要安装 CLI（**oc**） 来使用命令行界面与 OpenShift Container Platform 进行交互。您可在 Linux 、Windows 或 macOS 上安装 **oc**。



重要

如果安装了旧版本的 **oc**，则无法使用 OpenShift Container Platform 4.6 中的所有命令。下载并安装新版本的 **oc**。

1.1.7.1. 在 Linux 上安装 OpenShift CLI

您可以按照以下流程在 Linux 上安装 OpenShift CLI（**oc**）二进制文件。

流程

1. 进入到红帽客户门户网站上的 OpenShift Container Platform 下载页面。
2. 在 **Version** 下拉菜单中选择相应的版本。
3. 单击 **OpenShift v4.6 Linux 客户端** 条目旁边的 **Download Now**，再保存文件。
4. 解包存档：

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

5. 把 **oc** 二进制代码放到 **PATH** 中的目录中。
   执行以下命令可以查看当前的 **PATH** 设置：

```
$ echo $PATH
```

安装 OpenShift CLI 后，可以使用 **oc** 命令：

```
$ oc <command>
```

1.1.7.2. 在 Windows 上安装 OpenShift CLI

您可以按照以下流程在 Windows 上安装 OpenShift CLI（**oc**）二进制代码。

流程

1. 进入到红帽客户门户网站上的 OpenShift Container Platform 下载页面。
2. 在 **Version** 下拉菜单中选择相应的版本。
3. 单击 **OpenShift v4.6 Windows 客户端**条目旁边的 **Download Now**，再保存文件。
4. 使用 ZIP 程序解压存档。
5. 把 **oc** 二进制代码放到 **PATH** 中的目录中。
   要查看您的 **PATH**，请打开命令提示窗口并执行以下命令：

```
C:\> path
```

安装 OpenShift CLI 后，可以使用 **oc** 命令：

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

1.1.7.3. 在 macOS 上安装 OpenShift CLI

您可以按照以下流程在 macOS 上安装 OpenShift CLI（**oc**）二进制代码。

流程

1. 进入到红帽客户门户网站上的 OpenShift Container Platform 下载页面。
2. 在 **Version** 下拉菜单中选择相应的版本。
3. 单击 **OpenShift v4.6 MacOSX 客户端**条目旁边的 **Download Now**，再保存文件。
4. 解包和解压存档。
5. 将 **oc** 二进制文件移到 PATH 的目录中。
   要查看您的 **PATH**，打开一个终端窗口并执行以下命令：

   ```
   $ echo $PATH
   ```

安装 OpenShift CLI 后，可以使用 **oc** 命令：

```
$ oc <command>
```

1.1.8. 手动创建安装配置文件

对于使用用户置备的基础架构的 OpenShift Container Platform 安装，您必须手动生成安装配置文件。

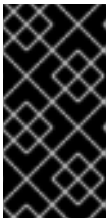
先决条件

- 获取 OpenShift Container Platform 安装程序和集群的访问令牌。

流程

1. 创建用来存储您所需的安装资产的安装目录：

   ```
   $ mkdir <installation_directory>
   ```

   重要

   您必须创建目录。一些安装信息，如 bootstrap X.509 证书，有较短的过期间隔，因此不要重复使用安装目录。如果要重复使用另一个集群安装中的个别文件，可以将其复制到您的目录中。但是，一些安装数据的文件名可能会在发行版本之间有所改变。从 OpenShift Container Platform 老版本中复制安装文件时要格外小心。

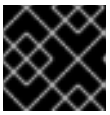
2. 自定义以下 **install-config.yaml** 文件模板，并将它保存到 **<installation_directory>** 中。

   注意

   此配置文件必须命名为 **install-config.yaml**。

3. 备份 **install-config.yaml** 文件，以便用于安装多个集群。



重要

install-config.yaml 文件会在安装过程的下一步骤中消耗掉。现在必须备份它。

1.1.8.1. 安装配置参数

在部署 OpenShift Container Platform 集群前，您可以提供参数值，以描述托管集群的云平台的帐户并选择性地自定义集群平台。在创建 **install-config.yaml** 安装配置文件时，您可以通过命令行来提供所需的参数的值。如果要自定义集群，可以修改 **install-config.yaml** 文件来提供关于平台的更多信息。

注意

安装之后，您无法修改 **install-config.yaml** 文件中的这些参数。

重要

openshift-install 命令不验证参数的字段名称。如果指定了不正确的名称，则不会创建相关的文件或对象，且不会报告错误。确保所有指定的参数的字段名称都正确。

1.1.8.1.1. 所需的配置参数

下表描述了所需的安装配置参数：

表 1.7. 所需的参数

参数	描述	值
apiVersion	**install-config.yaml** 内容的 API 版本。当前版本是 **v1**。安装程序还可能支持旧的 API 版本。	字符串
baseDomain	云供应商的基域。此基础域用于创建到 OpenShift Container Platform 集群组件的路由。集群的完整 DNS 名称是 **baseDomain** 和 **metadata.name** 参数值的组合，其格式为 **<metadata.name>.<baseDomain>**。	完全限定域名或子域名，如 **example.com**。
metadata	Kubernetes 资源 **ObjectMeta**，其中只消耗 **name** 参数。	对象
metadata.name	集群的名称。集群的 DNS 记录是 **{{.metadata.name}}.{{.baseDomain}}** 的子域。	小写字母,连字符(-)和句点(.)的字符串，如 **dev**。

参数	描述	值
platform	执行安装的具体平台配置：**aws**、**baremetal**、**azure**、**openstack**、**ovirt**、**vsphere**。有关 **platform.<platform>** 参数的额外信息，请参考下表来了解您的具体平台。	对象

pullSecret	从 Red Hat OpenShift Cluster Manager 获取 pull secret，验证从 Quay.io 等服务中下载 OpenShift Container Platform 组件的容器镜像。	<pre>{
		  "auths":{
		  "cloud.openshift.com":{
		  "auth":"b3Blb=",
		  "email":"you@example.com"
		  },
		  "quay.io":{
		  "auth":"b3Blb=",
		  "email":"you@example.com"
		  }
		  }
		}</pre>

1.1.8.1.2. 网络配置参数

您可以根据现有网络基础架构的要求自定义安装配置。例如，您可以扩展集群网络的 IP 地址块，或者提供不同于默认值的不同 IP 地址块。

只支持 IPv4 地址。

表 1.8. 网络参数

参数	描述	值
networking	集群网络的配置。	对象 **注意** 您不能在安装后修改 **networking** 对象指定的参数。
networking.networkType	要安装的集群网络供应商 Container Network Interface（CNI）插件。	**OpenShiftSDN** 或 **OVNKubernetes**。默认值为 **OpenShiftSDN**。

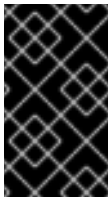
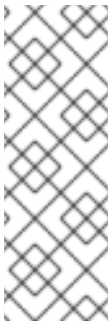
参数	描述	值
networking.clusterNetwork	pod 的 IP 地址块。 默认值为 **10.128.0.0/14**，主机前缀为 **/23**。 如果您指定多个 IP 地址块，则块不得互相重叠。	一个对象数组。例如： networking: clusterNetwork: - cidr: 10.128.0.0/14 hostPrefix: 23
networking.clusterNetwork.cidr	使用 **networking.clusterNetwork** 时需要此项。IP 地址块。 一个 IPv4 网络。	使用 CIDR 形式的 IP 地址块。IPv4 块的前缀长度介于 **0** 到 **32** 之间。
networking.clusterNetwork.hostPrefix	分配给每个单独节点的子网前缀长度。例如，如果 **hostPrefix** 设为 **23**，则每个节点从所给的 **cidr** 中分配一个 **/23** 子网。**hostPrefix** 值 **23** 提供 510（2^(32 - 23)- 2）个 pod IP 地址。	子网前缀。 默认值为 **23**。
networking.serviceNetwork	服务的 IP 地址块。默认值为 **172.30.0.0/16**。 OpenShift SDN 和 OVN-Kubernetes 网络供应商只支持服务网络的一个 IP 地址块。	CIDR 格式具有 IP 地址块的数组。例如： networking: serviceNetwork: - 172.30.0.0/16
networking.machineNetwork	机器的 IP 地址块。 如果您指定多个 IP 地址块，则块不得互相重叠。 如果您指定了多个 IP 内核参数，**machineNetwork.cidr** 值必须是主网络的 CIDR。	一个对象数组。例如： networking: machineNetwork: - cidr: 10.0.0.0/16
networking.machineNetwork.cidr	使用 **networking.machineNetwork** 时需要。IP 地址块。libvirt 以外的所有平台的默认值为 **10.0.0.0/16**。对于 libvirt，默认值为 **192.168.126.0/24**。	CIDR 表示法中的 IP 网络块。 例如：**10.0.0.0/16**。 **注意** 将 **networking.machineNetwork** 设置为与首选 NIC 所在的 CIDR 匹配。

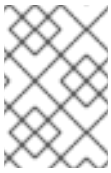
1.1.8.1.3. 可选配置参数

下表描述了可选安装配置参数：

表 1.9. 可选参数

参数	描述	值
additionalTrustBundle	添加到节点可信证书存储中的 PEM 编码 X.509 证书捆绑包。配置了代理时，也可以使用这个信任捆绑包。	字符串
compute	组成计算节点的机器的配置。	machine-pool 对象的数组。详情请查看以下"Machine-pool"表。
compute.architecture	决定池中机器的指令集合架构。目前不支持异构集群，因此所有池都必须指定相同的架构。有效值为 **amd64** （默认值）。	字符串
compute.hyperthreading	是否在计算机器上启用或禁用并发多线程或**超线程**。默认情况下，启用并发多线程以提高机器内核的性能。  **重要** 如果禁用并发多线程，请确保在容量规划时考虑到机器性能可能会显著降低的问题。	**Enabled** 或 **Disabled**
compute.name	使用 **compute** 时需要此值。机器池的名称。	**worker**
compute.platform	使用 **compute** 时需要此值。使用此参数指定托管 worker 机器的云供应商。此参数值必须与 **controlPlane.platform** 参数值匹配。	**aws**、**azure**、**gcp**、**openstack**、**ovirt**、**vsphere** 或 **{}**
compute.replicas	要置备的计算机器数量，也称为 worker 机器。	大于或等于 **2** 的正整数。默认值为 **3**。
controlPlane	组成 control plane 的机器的配置。	**MachinePool** 对象的数组。详情请查看以下"Machine-pool"表。
controlPlane.architecture	决定池中机器的指令集合架构。目前不支持异构集群，因此所有池都必须指定相同的架构。有效值为 **amd64** （默认值）。	字符串

参数	描述	值
controlPlane.hyperthreading	是否在 control plane 机器上启用或禁用并发多线程或**超线程**。默认情况下，启用并发多线程以提高机器内核的性能。  **重要** 如果禁用并发多线程，请确保在容量规划时考虑到机器性能可能会显著降低的问题。	**Enabled** 或 **Disabled**
controlPlane.name	使用 **controlPlane** 时需要。机器池的名称。	**master**
controlPlane.platform	使用 **controlPlane** 时需要。使用此参数指定托管 control plane 机器的云供应商。此参数值必须与 **compute.platform** 参数值匹配。	**aws**、**azure**、**gcp**、**openstack**、**ovirt**、**vsphere** 或 **{}**
controlPlane.replicas	要置备的 control plane 机器数量。	唯一支持的值是 **3**，它是默认值。
credentialsMode	Cloud Credential Operator（CCO）模式。如果没有指定任何模式，CCO 会动态地尝试决定提供的凭证的功能，在支持多个模式的平台上使用 mint 模式。  **注意** 不是所有 CCO 模式都支持所有云供应商。如需有关 CCO 模式的更多信息，请参阅*Red Hat Operator 参考指南*内容中的 *Cloud Credential Operator* 条目。	**Mint**、**Passthrough**、**Manual** 或空字符串(**""**)。

参数	描述	值
fips	启用或禁用 FIPS 模式。默认为 **false**（禁用）。如果启用了 FIPS 模式，运行 OpenShift Container Platform 的 Red Hat Enterprise Linux CoreOS (RHCOS) 机器会绕过默认的 Kubernetes 加密套件，并使用由 RHCOS 提供的加密模块。  **重要** 只有在 **x86_64** 架构中的 OpenShift Container Platform 部署支持 FIPS 验证的/Modules in Process 加密库。  **注意** 如果使用 Azure File 存储，则无法启用 FIPS 模式。	**false** 或 **true**
imageContentSources	release-image 内容的源和仓库。	对象数组。包括一个 **source** 以及可选的 **mirrors**，如下表所示。
imageContentSources.source	使用 **imageContentSources** 时需要。指定用户在镜像拉取规格中引用的仓库。	字符串
imageContentSources.mirrors	指定可能还包含同一镜像的一个或多个仓库。	字符串数组
publish	如何发布或公开集群的面向用户的端点，如 Kubernetes API、OpenShift 路由。	**Internal** 或 **External**。默认值为 **External**。 在非云平台上不支持将此字段设置为 **Internal**。  **重要** 如果将字段的值设为 **Internal**，集群将无法运行。如需更多信息，请参阅 BZ#1953035。

参数	描述	值
sshKey	用于验证集群机器访问的 SSH 密钥或密钥。 **注意** 对于您要在其上执行安装调试或灾难恢复的生产环境 OpenShift Container Platform 集群，请指定 **ssh-agent** 进程使用的 SSH 密钥。	一个或多个密钥。例如： sshKey: <key1> <key2> <key3>

1.1.8.2. IBM Z 的 install-config.yaml 文件示例

您可以自定义 **install-config.yaml** 文件，以指定有关 OpenShift Container Platform 集群平台的更多信息，或修改所需参数的值。

```
apiVersion: v1
baseDomain: example.com (1)
compute: (2)
- hyperthreading: Enabled (3)
  name: worker
  replicas: 0 (4)
  architecture : s390x
controlPlane: (5)
  hyperthreading: Enabled (6)
  name: master
  replicas: 3 (7)
  architecture : s390x
metadata:
  name: test (8)
networking:
  clusterNetwork:
  - cidr: 10.128.0.0/14 (9)
    hostPrefix: 23 (10)
  networkType: OpenShiftSDN
  serviceNetwork: (11)
  - 172.30.0.0/16
platform:
  none: {} (12)
fips: false (13)
pullSecret: '{"auths": ...}' (14)
sshKey: 'ssh-ed25519 AAAA...' (15)
```

(1) 集群的基域。所有 DNS 记录都必须是这个基域的子域，并包含集群名称。

(2) (5)

controlPlane 部分是一个单映射，但 **compute** 部分是一系列映射。为满足不同数据结构的要求，**compute** 部分的第一行必须以连字符 **-** 开头，**controlPlane** 部分的第一行则不可以连字符开

3 6 是否要启用或禁用并发多线程（SMT）或**超线程**。默认情况下，启用 SMT 可提高机器内核的性能。您可以通过将参数值设为 **Disabled** 来禁用。如果禁用 SMT，则必须在所有集群机器中禁用它，其中包括 control plane 和计算机器。

注意

默认启用并发多线程（SMT）。如果在 BIOS 设置中没有启用 SMT，**hyperthreading** 参数不会起作用。

重要

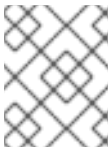
如果您禁用 **hyperthreading**（无论是在 BIOS 中还是在 **install-config.yaml** 中），请确保您对可能会造成的机器性能显著降低的情况有所考虑。

4 **replicas** 参数的值必须设置为 **0**。此参数控制集群为您创建和管理的 worker 数量，使用用户置备的基础架构时集群不会执行这些功能。在完成 OpenShift Container Platform 安装前，您必须手动为集群部署 worker 机器。

7 您添加到集群的 control plane 机器数量。由于集群将这个值用作集群中 etcd 端点的数量，因此该值必须与您部署的 control plane 机器数量匹配。

8 您在 DNS 记录中指定的集群名称。

9 从中分配 pod IP 地址的 IP 地址块。此块不得与现有的物理网络重叠。这些 IP 地址用于 pod 网络。如果您需要从外部网络访问 pod，请配置负载均衡器和路由器来管理流量。



注意

类 E CIDR 范围保留给以后使用。要使用 Class E CIDR 范围，您必须确保您的网络环境接受 Class E CIDR 范围内的 IP 地址。

10 分配给每个单独节点的子网前缀长度。例如，如果 **hostPrefix** 设为 **23**，则每个节点从所给的 **cidr** 中分配一个 **/23** 子网，这样就能有 510 (2^(32 - 23) - 2) 个 Pod IP 地址。如果您需要从外部网络访问节点，请配置负载均衡器和路由器来管理流量。

11 用于服务 IP 地址的 IP 地址池。您只能输入一个 IP 地址池。此块不得与现有的物理网络重叠。如果您需要从外部网络访问服务，请配置负载均衡器和路由器来管理流量。

12 您必须将平台设置为 **none**。您无法为 IBM Z 基础架构提供额外的平台配置变量。

13 是否启用或禁用 FIPS 模式。默认情况下不启用 FIPS 模式。如果启用了 FIPS 模式，运行 OpenShift Container Platform 的 Red Hat Enterprise Linux CoreOS (RHCOS) 机器会绕过默认的 Kubernetes 加密套件，并使用由 RHCOS 提供的加密模块。

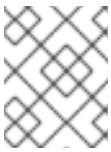
重要

只有在 **x86_64** 架构中的 OpenShift Container Platform 部署支持 FIPS 验证的/Modules in Process 加密库。

14

Red Hat OpenShift Cluster Manager 中的 pull secret 。通过此 pull secret，您可以进行所含授权机构提供的服务的身份验证，这些服务包括为 OpenShift Container Platform 组件提供容器镜像的

15 Red Hat Enterprise Linux CoreOS (RHCOS) 中 **core** 用户的默认 SSH 密钥的公钥部分。



注意

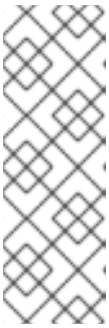
对于您要在其上执行安装调试或灾难恢复的生产环境 OpenShift Container Platform 集群，请指定 **ssh-agent** 进程使用的 SSH 密钥。

1.1.9. 在安装过程中配置集群范围代理

生产环境可能会拒绝直接访问互联网，而是提供 HTTP 或 HTTPS 代理。您可以通过在 **install-config.yaml** 文件中配置代理设置，将新的 OpenShift Container Platform 集群配置为使用代理。

先决条件

- 您有一个现有的 **install-config.yaml** 文件。
- 您检查了集群需要访问的站点，并决定是否需要绕过代理。默认情况下代理所有集群出口流量，包括对托管云供应商 API 的调用。您需要将站点添加到 **Proxy** 对象的 **spec.noProxy** 字段来绕过代理。



注意

Proxy 对象 **status.noProxy** 字段使用安装配置中的 **networking.machineNetwork[].cidr**、**networking.clusterNetwork[].cidr** 和 **networking.serviceNetwork[]** 字段的值填充。

对于在 Amazon Web Services(AWS)、Google Cloud Platform(GCP)、Microsoft Azure 和 Red Hat OpenStack Platform(RHOSP)上安装, **Proxy** 对象 **status.noProxy** 字段也会使用实例元数据端点填充(**169.254.169.254**)。

流程

1. 编辑 **install-config.yaml** 文件并添加代理设置。例如：

```
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-----
...
```

1 用于创建集群外 HTTP 连接的代理 URL。URL 必须是 **http**。

2 用于创建集群外 HTTPS 连接的代理 URL。

3 要排除在代理中的目标域名、IP 地址或其他网络 CIDR 的逗号分隔列表。在域前面加 **.** 来仅匹配子域。例如： **.y.com** 匹配 **x.y.com**，但不匹配 **y.com**。使用 ***** 绕过所有目的地的代

4 如果提供，安装程序会在 **openshift-config** 命名空间中生成名为 **user-ca-bundle** 的配置映射，以容纳额外的 CA 证书。如果您提供 **additionalTrustBundle** 和至少一个代理设置，**Proxy** 对象会被配置为引用 **trustedCA** 字段中的 **user-ca-bundle** 配置映射。然后，Cluster Network Operator 会创建一个 **trusted-ca-bundle** 配置映射，将 **trustedCA** 参数指定的值与 RHCOS 信任捆绑包合并。**additionalTrustBundle** 字段是必需的，除非代理的身份证书由来自 RHCOS 信任捆绑包的颁发机构签名。

注意

安装程序不支持代理的 **readinessEndpoints** 字段。

2. 保存该文件，并在安装 OpenShift Container Platform 时引用。

安装程序会创建一个名为 **cluster** 的集群范围代理，该代理使用提供的 **install-config.yaml** 文件中的代理设置。如果没有提供代理设置，仍然会创建一个 **cluster Proxy** 对象，但它会有一个空 **spec**。

注意

只支持名为 **cluster** 的 **Proxy** 对象，且无法创建额外的代理。

1.1.10. 创建 Kubernetes 清单和 Ignition 配置文件

由于您必须修改一些集群定义文件并要手动启动集群机器，因此您必须生成 Kubernetes 清单和 Ignition 配置文件，集群需要这两项来创建其机器。

安装配置文件转换为 Kubernetes 清单。清单嵌套到 Ignition 配置文件中，稍后用于创建集群。

重要

- 安装程序生成的 Ignition 配置文件包含在 24 小时后过期的证书，然后在过期时进行续订。如果在更新证书前关闭集群，且集群在 24 小时后重启，集群会自动恢复过期的证书。一个例外情况是，您需要手动批准待处理的 **node-bootstrapper** 证书签名请求（CSR）来恢复 kubelet 证书。如需更多信息，请参阅 *从过期的 control plane 证书中恢复*的文档。
- 建议您在生成 12 小时后使用 Ignition 配置文件，因为集群安装后 24 小时证书从 16 小时轮转至 22 小时。通过在 12 小时内使用 Ignition 配置文件，您可以避免在安装过程中运行证书更新时避免安装失败。

先决条件

- 已获得 OpenShift Container Platform 安装程序。
- 已创建 **install-config.yaml** 安装配置文件。

流程

1. 切换到包含安装程序的目录，并为集群生成 Kubernetes 清单：

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

1 对于 **<installation_directory>**，请指定含有您创建的 **install-config.yaml** 文件的安装目录。

2. 检查 **<installation_directory>/manifests/cluster-scheduler-02-config.yml** Kubernetes 清单文件中的 **mastersSchedulable** 参数是否已设置为 **false**。此设置可防止在 control plane 机器上调度 pod:

   a. 打开 **<installation_directory>/manifests/cluster-scheduler-02-config.yml** 文件。

   b. 找到 **mastersSchedulable** 参数并确保它被设置为 **false**。

   c. 保存并退出文件。

3. 要创建 Ignition 配置文件，从包含安装程序的目录运行以下命令：

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

1 对于 **<installation_directory>**，请指定相同的安装目录。

该目录中将生成以下文件：

```
.
├── auth
│   ├── kubeadmin-password
│   └── kubeconfig
├── bootstrap.ign
├── master.ign
├── metadata.json
└── worker.ign
```

1.1.11. 创建 Red Hat Enterprise Linux CoreOS (RHCOS) 机器

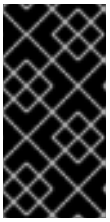
在您置备的 IBM Z 环境中安装集群前，您必须在 z/VM 虚拟机上安装 RHCOS 以便集群使用。完成以下步骤以创建机器。

先决条件

- 在置备机器中运行 FTP 服务器，您创建的机器需要可以访问这个 FTP 服务器。

流程

1. 在您置备的机器上登录到 Linux。
2. 从 RHCOS 镜像镜像 获取 Red Hat Enterprise Linux CoreOS（RHCOS）内核、initramfs 和 rootfs 文件。



重要

RHCOS 镜像可能不会随着 OpenShift Container Platform 的每一发行版本都有改变。您必须下载最高版本的镜像，其版本号应小于或等于您安装的 OpenShift Container Platform 版本。只使用以下流程中描述的适当内核、initramfs 和 rootfs 工件。

文件名包含 OpenShift Container Platform 版本号。它们类似以下示例：

- kernel: **rhcos-<version>-live-kernel-<architecture>**
- initramfs: **rhcos-<version>-live-initramfs.<architecture>.img**
- rootfs: **rhcos-<version>-live-rootfs.<architecture>.img**

注意

FCP 和 DASD 的 rootfs 镜像是相同的。

3. 创建参数文件。以下参数特定于特定虚拟机：
   - 对于 **coreos.inst.install_dev=**，请为 DASD 安装指定 **dasda**，或者为 FCP 指定 **sda**。请注意 FCP 需要 **zfcp.allow_lun_scan=0**。
   - 对于 **rd.dasd=**，请指定要安装 RHCOS 的 DASD。
   - **rd.zfcp=<adapter>,<wwpn>,<lun>** 指定要在其中安装 RHCOS 的 FCP 磁盘。
   - 对于 **ip=**，请指定以下七项：
     i. 机器的 IP 地址。
     ii. 一个空字符串。
     iii. 网关
     iv. 子网掩码。
     v. **hostname.domainname**格式的机器主机和域名。省略这个值会让 RHCOS 来决定这个值。
     vi. 网络接口名称。省略这个值会让 RHCOS 来决定这个值。
     vii. 如果使用静态 IP 地址，则为一个空字符串。
   - 对于 **coreos.inst.ignition_url=**，为机器角色指定 Ignition 文件。使用 **bootstrap.ign**、**master.ign** 或 **worker.ign**。只支持 HTTP 和 HTTPS 协议。
   - 对于 **coreos.live.rootfs_url=**，为您引导的内核和 initramfs 指定匹配的 rootfs 工件。只支持 HTTP 和 HTTPS 协议。
   - 所有其他参数都可以保留。
     bootstrap 机器的实例参数文件（**bootstrap-0.parm**）如下：

```
rd.neednet=1 \
console=ttysclp0 \
coreos.inst.install_dev=dasda \
coreos.live.rootfs_url=http://cl1.provide.example.com:8080/assets/rhcos-live-
rootfs.s390x.img \
coreos.inst.ignition_url=http://cl1.provide.example.com:8080/ignition/bootstrap.ign \
ip=172.18.78.2::172.18.78.1:255.255.255.0:::none nameserver=172.18.78.1 \
```

```
rd.znet=qeth,0.0.bdf0,0.0.bdf1,0.0.bdf2,layer2=1,portno=0 \
zfcp.allow_lun_scan=0 \
rd.dasd=0.0.3490
```

将参数文件中的所有选项写为一行，并确保您没有换行符。

4. 将 initramfs 、内核、参数文件和 RHCOS 镜像传送到 z/VM 中，例如使用 FTP。有关如何使用 FTP 传输文件并从虚拟 reader 引导的详情，请参考 在 Z/VM 中安装。

5. 将文件 punch 到 z/VM 虚拟机的虚拟 reader，即成为 bootstrap 节点。
请参阅 IBM 文档中的 PUNCH。

   提示

   您可以使用 CP PUNCH 命令（如果是 Linux，使用 **vmur** 命令）在两个 z/VM 虚拟机间传输文件。

6. 在 bootstrap 机器中登录到 CMS。

7. 从 reader IPL bootstrap 机器：

   ```
   $ ipl c
   ```

   请参阅 IBM 文档中的 IPL。

8. 对集群中的其他机器重复此步骤。

1.1.11.1. 高级 RHCOS 安装参考

本节演示了网络配置和其他高级选项，允许您修改 Red Hat Enterprise Linux CoreOS（RHCOS）手动安装过程。下表描述了您可以与 RHCOS live installer 和 **coreos-installer** 命令一起使用的内核参数和命令行选项。

RHCOS 启动提示下的路由和绑定选项
如果从 ISO 镜像安装 RHCOS，您可以在引导该镜像时手动添加内核参数以配置节点的网络。如果没有使用网络参数，则安装默认为使用 DHCP。

重要

添加网络参数时，还必须添加 **rd.neednet=1** 内核参数。

下表描述了如何为实时 ISO 安装使用 **ip=** 、 **nameserver=** 和 **bond=** 内核参数。

注意

在添加内核参数时顺序非常重要： **ip=**，**nameserver=**，然后 **bond=**。

ISO 的路由和绑定选项

下表提供了配置 Red Hat Enterprise Linux CoreOS（RHCOS）节点网络的示例。这些是在系统引导过程中传递给 **dracut** 工具的网络选项。有关 **dracut** 支持的网络选项的详情，请参考 **dracut.cmdline** 手册页。

描述	例子
要配置一个 IP 地址，可以使用 DHCP(**ip=dhcp**)或者设置单独的静态 IP 地址(**ip=<host_ip>**)。然后在每个节点上指定 DNS 服务器 IP 地址(**nameserver=<dns_ip>**)。这个示例设置： • 节点的 IP 地址为 **10.10.10.2** • 网关地址为 **10.10.10.254** • 子网掩码为 **255.255.255.0** • 主机名为 **core0.example.com** • DNS 服务器地址为 **4.4.4.41**	ip=10.10.10.2::10.10.10.254:255.255.255.0:core0.example.com:enp1s0:none nameserver=4.4.4.41
通过指定多个 **ip=** 条目来指定多个网络接口。	ip=10.10.10.2::10.10.10.254:255.255.255.0:core0.example.com:enp1s0:none ip=10.10.10.3::10.10.10.254:255.255.255.0:core0.example.com:enp2s0:none
可选：您可以通过设置一个 **rd.route=** 值来配置到额外网络的路由。 如果额外网络网关与主要网络网关不同，则默认网关必须是主要网络网关。	配置默认网关： ip=::10.10.10.254:::: 为额外网络配置路由： rd.route=20.20.20.0/24:20.20.20.254:enp2s0
在单一接口中禁用 DHCP，比如当有两个或者多个网络接口时，且只有一个接口被使用。	ip=10.10.10.2::10.10.10.254:255.255.255.0:core0.example.com:enp1s0:none ip=::::core0.example.com:enp2s0:none
您可以将系统中 DHCP 和静态 IP 配置与多个网络接口结合在一起。	ip=enp1s0:dhcp ip=10.10.10.2::10.10.10.254:255.255.255.0:core0.example.com:enp2s0:none

描述	例子
可选： 您可以使用 **vlan=** 参数在单独的接口上配置 VLAN。	在网络接口中配置 VLAN 并使用静态 IP 地址： ``` ip=10.10.10.2::10.10.10.254:255.255.255.0:core0.example.com:enp2s0.100:none vlan=enp2s0.100:enp2s0 ``` 在网络接口中配置 VLAN 并使用 DHCP： ``` ip=enp2s0.100:dhcp vlan=enp2s0.100:enp2s0 ```
您可以为每个服务器添加一个 **nameserver=** 条目来提供多个 DNS 服务器。	``` nameserver=1.1.1.1 nameserver=8.8.8.8 ```
可选：使用 **bond=** 选项支持将多个网络接口绑定到一个接口。在这两个示例中： - 配置绑定接口的语法为：**bond=name[:network_interfaces][:options]** - *name* 是绑定设备名称（**bond0**），*network_interfaces* 代表用逗号分开的物理（以太网）接口(**em1,em2**)的列表，*options* 是用逗号分开的绑定选项列表。输入 **modinfo bonding** 查看可用选项。 - 当使用 **bond=** 创建绑定接口时，您必须指定如何分配 IP 地址以及绑定接口的其他信息。	要将绑定的接口配置为使用 DHCP，请将绑定的 IP 地址设置为 **dhcp**。例如： ``` bond=bond0:em1,em2:mode=active-backup ip=bond0:dhcp ``` 要将绑定接口配置为使用静态 IP 地址，请输入您需要的特定 IP 地址以及相关信息。例如： ``` bond=bond0:em1,em2:mode=active-backup ip=10.10.10.2::10.10.10.254:255.255.255.0:core0.example.com:bond0:none ```
可选： 您可以使用 **vlan=** 参数在绑定接口上配置 VLAN。	使用 VLAN 配置绑定接口并使用 DHCP： ``` ip=bond0.100:dhcp bond=bond0:em1,em2:mode=active-backup vlan=bond0.100:bond0 ``` 使用 VLAN 配置绑定接口，并使用静态 IP 地址： ``` ip=10.10.10.2::10.10.10.254:255.255.255.0:core0.example.com:bond0.100:none bond=bond0:em1,em2:mode=active-backup vlan=bond0.100:bond0 ```

描述	例子

可选：使用 **team=** 参数将网络团队用作绑定的替代选择。在本例中：
• 配置组接口的语法为：**team=name[:network_interfaces]** *name* 是团队设备名称（**team0**），*network_interfaces* 代表用逗号分开的物理（以太网）接口（**em1、em2**）。
 	
注意
当 RHCOS 切换到即将发布的 RHEL 版本时，团队计划被弃用。如需更多信息，请参阅 Red Hat 知识库文章。
	配置网络团队：
	`team=team0:em1,em2`
	`ip=team0:dhcp`

1.1.12. 创建集群

要创建 OpenShift Container Platform 集群，请等待您通过安装程序生成的 Ignition 配置文件所置备的机器上完成 bootstrap 过程。

先决条件

- 为集群创建所需的基础架构。
- 已获得安装程序并为集群生成了 Ignition 配置文件。
- 已使用 Ignition 配置文件为集群创建 RHCOS 机器。
- 您的机器可直接访问互联网，或者可以使用 HTTP 或 HTTPS 代理。

流程

1. 监控 bootstrap 过程：

```
$ ./openshift-install --dir <installation_directory> wait-for bootstrap-complete \ 1
    --log-level=info 2
```

1 对于 **<installation_directory>**，请指定安装文件保存到的目录的路径。

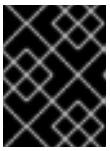
2 要查看不同的安装详情，请指定 **warn**、**debug** 或 **error**，而不要指定 **info**。

输出示例

```
INFO Waiting up to 30m0s for the Kubernetes API at https://api.test.example.com:6443...
INFO API v1.19.0 up
INFO Waiting up to 30m0s for bootstrapping to complete...
INFO It is now safe to remove the bootstrap resources
```

Kubernetes API 服务器提示已在 control plane 机器上完成 bootstrap 时，命令运行成功。

2. bootstrap 过程完成后，请从负载均衡器中删除 bootstrap 机器。



重要

此时您必须从负载均衡器中删除 bootstrap 机器。您还可以删除或重新格式化机器本身。

1.1.13. 使用 CLI 登录到集群

您可以通过导出集群 **kubeconfig** 文件，以默认系统用户身份登录集群。**kubeconfig** 文件包含关于集群的信息，供 CLI 用于将客户端连接到正确集群和 API 服务器。该文件特只适用于一个特定的集群，在 OpenShift Container Platform 安装过程中创建。

先决条件

- 已部署了 OpenShift Container Platform 集群。
- 已安装 **oc** CLI。

流程

1. 导出 **kubeadmin** 凭证：

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

1 对于 **<installation_directory>**，请指定安装文件保存到的目录的路径。

2. 使用导出的配置，验证能否成功运行 **oc** 命令：

```
$ oc whoami
```

输出示例

```
system:admin
```

1.1.14. 批准机器的证书签名请求

将机器添加到集群时，会为您添加的每台机器生成两个待处理证书签名请求（CSR）。您必须确认这些 CSR 已获得批准，或根据需要自行批准。客户端请求必须首先被批准，然后是服务器请求。

先决条件

- 您已将机器添加到集群中。

流程

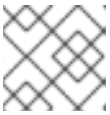
1. 确认集群可以识别这些机器：

```
$ oc get nodes
```

输出示例

```
NAME     STATUS   ROLES   AGE  VERSION
master-0 Ready    master  63m  v1.19.0
master-1 Ready    master  63m  v1.19.0
master-2 Ready    master  64m  v1.19.0
```

输出将列出您创建的所有机器。



注意

在一些 CSR 被批准前，以上输出可能不包括计算节点（也称为 worker 节点）。

2. 检查待处理的 CSR，并确保可以看到添加到集群中的每台机器都有 **Pending** 或 **Approved** 状态的客户端请求：

```
$ oc get csr
```

输出示例

```
NAME        AGE   REQUESTOR                                   CONDITION
csr-mddf5   20m   system:node:master-01.example.com   Approved,Issued
csr-z5rln   16m   system:node:worker-21.example.com   Approved,Issued
```

3. 如果 CSR 没有获得批准，请在所添加机器的所有待处理 CSR 都处于 **Pending** 状态后，为您的集群机器批准这些 CSR：

注意

由于 CSR 会自动轮转，因此请在将机器添加到集群后一小时内批准您的 CSR。如果没有在一小时内批准，证书将会轮转，每个节点将会存在多个证书。您必须批准所有这些证书。批准客户端 CSR 后， Kubelet 为服务证书创建辅助 CSR，这需要手动批准。然后，如果 Kubelet 请求具有相同参数的新证书，则 **machine-approver** 会自动批准后续服务证书续订请求。

注意

对于在未启用机器 API 的平台中运行的集群，如裸机和其他用户置备的基础架构，必须采用一种方法自动批准 kubelet 提供证书请求（CSR）。如果没有批准请求，则 **oc exec**、**oc rsh** 和 **oc logs** 命令将无法成功，因为 API 服务器连接到 kubelet 时需要服务证书。与 Kubelet 端点联系的任何操作都需要此证书批准。这个方法必须监视新的 CSR，确认 CSR 由 **system:node** 或 **system:admin** 组中的 **node-bootstrapper** 服务帐户提交，并确认节点的身份。

- 若要单独批准，请对每个有效的 CSR 运行以下命令：

```
$ oc adm certificate approve <csr_name> 1
```

1 **<csr_name>** 是当前 CSR 列表中 CSR 的名称。

- 要批准所有待处理的 CSR，请运行以下命令：

```
$ 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
```

注意

在有些 CSR 被批准前，一些 Operator 可能无法使用。

4. 现在，您的客户端请求已被批准，您必须查看添加到集群中的每台机器的服务器请求：

```
$ oc get csr
```

输出示例

```
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. 如果剩余的 CSR 没有被批准，且处于 **Pending** 状态，请批准集群机器的 CSR：

- 若要单独批准，请对每个有效的 CSR 运行以下命令：

```
$ oc adm certificate approve <csr_name> 1
```

1 **<csr_name>** 是当前 CSR 列表中 CSR 的名称。

- 要批准所有待处理的 CSR，请运行以下命令：

```
$ oc get csr -o go-template='{{range .items}}{{if not .status}}{{.metadata.name}}{{"\n"}}
{{end}}{{end}}' | xargs oc adm certificate approve
```

6. 批准所有客户端和服务器 CSR 后，器将处于 **Ready** 状态。运行以下命令验证：

```
$ oc get nodes
```

输出示例

```
NAME      STATUS    ROLES   AGE  VERSION
master-0  Ready     master  73m  v1.20.0
master-1  Ready     master  73m  v1.20.0
master-2  Ready     master  74m  v1.20.0
worker-0  Ready     worker  11m  v1.20.0
worker-1  Ready     worker  11m  v1.20.0
```

注意

批准服务器 CSR 后可能需要几分钟时间让机器转换为 **Ready** 状态。

其他信息

- 如需有关 CSR 的更多信息，请参阅证书签名请求。

1.1.15. 初始 Operator 配置

在 control plane 初始化后，您必须立即配置一些 Operator 以便它们都可用。

先决条件

- 您的 control plane 已初始化。

流程

1. 观察集群组件上线：

```
$ watch -n5 oc get clusteroperators
```

输出示例

```
NAME                                       VERSION AVAILABLE   PROGRESSING   DEGRADED
SINCE
authentication                             4.6.0   True        False         False      3h56m
cloud-credential                           4.6.0   True        False         False      29h
cluster-autoscaler                         4.6.0   True        False         False      29h
config-operator                            4.6.0   True        False         False      6h39m
console                                    4.6.0   True        False         False      3h59m
csi-snapshot-controller                    4.6.0   True        False         False      4h12m
dns                                        4.6.0   True        False         False      4h15m
etcd                                       4.6.0   True        False         False      29h
image-registry                             4.6.0   True        False         False      3h59m
ingress                                    4.6.0   True        False         False      4h30m
insights                                   4.6.0   True        False         False      29h
kube-apiserver                             4.6.0   True        False         False      29h
kube-controller-manager                    4.6.0   True        False         False      29h
kube-scheduler                             4.6.0   True        False         False      29h
kube-storage-version-migrator              4.6.0   True        False         False      4h2m
machine-api                                4.6.0   True        False         False      29h
machine-approver                           4.6.0   True        False         False      6h34m
machine-config                             4.6.0   True        False         False      3h56m
marketplace                                4.6.0   True        False         False      4h2m
monitoring                                 4.6.0   True        False         False      6h31m
network                                    4.6.0   True        False         False      29h
node-tuning                                4.6.0   True        False         False      4h30m
openshift-apiserver                        4.6.0   True        False         False      3h56m
openshift-controller-manager               4.6.0   True        False         False      4h36m
openshift-samples                          4.6.0   True        False         False      4h30m
operator-lifecycle-manager                 4.6.0   True        False         False      29h
operator-lifecycle-manager-catalog         4.6.0   True        False         False      29h
operator-lifecycle-manager-packageserver   4.6.0   True        False         False      3h59m
service-ca                                 4.6.0   True        False         False      29h
storage                                    4.6.0   True        False         False      4h30m
```

2. 配置不可用的 Operator。

1.1.15.1. 镜像 registry 存储配置

对于不提供默认存储的平台，Image Registry Operator 最初将不可用。安装后，您必须配置 registry 使用的存储，这样 Registry Operator 才可用。

示配置生产集群所需的持久性卷的说明。如果适用，显示有关将空目录配置为存储位置的说明，该位置只可用于非生产集群。

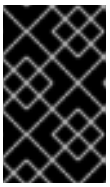
另外还提供了在升级过程中使用 **Recreate** rollout 策略来允许镜像 registry 使用块存储类型的说明。

1.1.15.1.1. 为 IBM Z 配置 registry 存储

作为集群管理员，在安装后需要配置 registry 来使用存储。

先决条件

- 具有 Cluster Administrator 权限
- IBM Z 上的集群。
- 为集群置备的持久性存储。



重要

如果您只有一个副本，OpenShift Container Platform 支持对镜像 registry 存储的 **ReadWriteOnce** 访问。要部署支持高可用性的、带有两个或多个副本的镜像 registry，需要 **ReadWriteMany** 访问设置。

- 必须具有 100Gi 容量。

流程

1. 为了配置 registry 使用存储，需要修改 **configs.imageregistry/cluster** 资源中的 **spec.storage.pvc** 。

注意

使用共享存储时，请查看您的安全设置以防止被外部访问。

2. 验证您没有 registry pod:

```
$ oc get pod -n openshift-image-registry
```

注意

如果存储类型为 **emptyDIR**，则副本数不能超过 **1**。

3. 检查 registry 配置：

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

输出示例

-

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

将 **claim** 字段留空以允许自动创建一个 **image-registry-storage** PVC。

4. 检查 **clusteroperator** 的状态：

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

5. 确保您的 registry 设置为 manage，以启用镜像的构建和推送。

- 运行：

```
$ oc edit configs.imageregistry/cluster
```

然后将行改

```
managementState: Removed
```

为

```
managementState: Managed
```

1.1.15.1.2. 在非生产集群中配置镜像 registry 存储

您必须为 Image Registry Operator 配置存储。对于非生产集群，您可以将镜像 registry 设置为空目录。如果您这样做，重启 registry 后会丢失所有镜像。

流程

- 将镜像 registry 存储设置为空目录：

```
$ oc patch configs.imageregistry.operator.openshift.io cluster --type merge --patch '{"spec":{"storage":{"emptyDir":{}}}}'
```

警告

仅可为非生产集群配置这个选项。

如果在 Image Registry Operator 初始化其组件前运行此命令，**oc patch** 命令会失败并显示以下错误：

```
Error from server (NotFound): configs.imageregistry.operator.openshift.io "cluster" not found
```

等待几分钟，然后再次运行该命令。

1.1.16. 在用户置备的基础架构上完成安装

完成 Operator 配置后，可以在您提供的基础架构上完成集群安装。

先决条件

- 您的 control plane 已初始化。
- 已完成初始 Operator 配置。

流程

1. 使用以下命令确认所有集群组件都已在线：

```
$ watch -n5 oc get clusteroperators
```

输出示例

```
NAME                                       VERSION AVAILABLE   PROGRESSING   DEGRADED
SINCE
authentication                             4.6.0   True        False         False      3h56m
cloud-credential                           4.6.0   True        False         False      29h
cluster-autoscaler                         4.6.0   True        False         False      29h
config-operator                            4.6.0   True        False         False      6h39m
console                                    4.6.0   True        False         False      3h59m
csi-snapshot-controller                    4.6.0   True        False         False      4h12m
dns                                        4.6.0   True        False         False      4h15m
etcd                                       4.6.0   True        False         False      29h
image-registry                             4.6.0   True        False         False      3h59m
ingress                                    4.6.0   True        False         False      4h30m
insights                                   4.6.0   True        False         False      29h
kube-apiserver                             4.6.0   True        False         False      29h
kube-controller-manager                    4.6.0   True        False         False      29h
kube-scheduler                             4.6.0   True        False         False      29h
kube-storage-version-migrator              4.6.0   True        False         False      4h2m
machine-api                                4.6.0   True        False         False      29h
machine-approver                           4.6.0   True        False         False      6h34m
machine-config                             4.6.0   True        False         False      3h56m
marketplace                                4.6.0   True        False         False      4h2m
monitoring                                 4.6.0   True        False         False      6h31m
network                                    4.6.0   True        False         False      29h
node-tuning                                4.6.0   True        False         False      4h30m
openshift-apiserver                        4.6.0   True        False         False      3h56m
openshift-controller-manager               4.6.0   True        False         False      4h36m
openshift-samples                          4.6.0   True        False         False      4h30m
operator-lifecycle-manager                 4.6.0   True        False         False      29h
operator-lifecycle-manager-catalog         4.6.0   True        False         False      29h
operator-lifecycle-manager-packageserver   4.6.0   True        False         False      3h59m
service-ca                                 4.6.0   True        False         False      29h
storage                                    4.6.0   True        False         False      4h30m
```

或者，通过以下命令，如果所有集群都可用您会接到通知。它还检索并显示凭证：

```
$ ./openshift-install --dir <installation_directory> wait-for install-complete 1
```

1 对于 **<installation_directory>**，请指定安装文件保存到的目录的路径。

输出示例

```
INFO Waiting up to 30m0s for the cluster to initialize...
```

Cluster Version Operator 完成从 Kubernetes API 服务器部署 OpenShift Container Platform 集群时，命令运行成功。

重要

- 安装程序生成的 Ignition 配置文件包含在 24 小时后过期的证书，然后在过期时进行续订。如果在更新证书前关闭集群，且集群在 24 小时后重启，集群会自动恢复过期的证书。一个例外情况是，您需要手动批准待处理的 **node-bootstrapper** 证书签名请求（CSR）来恢复 kubelet 证书。如需更多信息，请参阅*从过期的 control plane 证书中恢复*的文档。
- 建议您在生成 12 小时后使用 Ignition 配置文件，因为集群安装后 24 小时证书从 16 小时轮转至 22 小时。通过在 12 小时内使用 Ignition 配置文件，您可以避免在安装过程中运行证书更新时避免安装失败。

2. 确认 Kubernetes API 服务器正在与 pod 通信。

   a. 要查看所有 pod 的列表，请使用以下命令：

   ```
   $ oc get pods --all-namespaces
   ```

   输出示例

   ```
   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. 使用以下命令，查看上一命令的输出中所列 pod 的日志：

   ```
   $ oc logs <pod_name> -n <namespace>
   ```
   1

   1 指定 pod 名称和命名空间，如上一命令的输出中所示。

   如果 pod 日志显示，Kubernetes API 服务器可以与集群机器通信。

1.1.17. OpenShift Container Platform 的 Telemetry 访问

在 OpenShift Container Platform 4.6 中，默认运行的 Telemetry 服务提供有关集群健康状况和成功更新的指标，需要访问互联网。如果您的集群连接到互联网，Telemetry 会自动运行，而且集群会注册到 OpenShift Cluster Manager。

确认 OpenShift Cluster Manager 清单正确后，可以由 Telemetry 自动维护，也可以使用 OpenShift Cluster Manager 手动维护，使用订阅监控来跟踪帐户或多集群级别的 OpenShift Container Platform 订阅。

其他资源

- 有关 Telemetry 服务的更多信息，请参阅关于远程健康监控。

1.1.18. 收集调试信息

您可以收集有助于在 IBM Z 中安装 OpenShift Container Platform 时对特定问题进行故障排除和调试的调试信息。

先决条件

- 安装了 **oc** CLI 工具

流程

1. 登录到集群：

```
$ oc login
```

2. 在您要收集硬件信息的节点中，启动一个调试容器：

```
$ oc debug node/<nodename>
```

3. 进入 **/host** 文件系统并启动 **toolbox**:

```
$ chroot /host
$ toolbox
```

4. 收集 **dbginfo** 数据：

```
$ dbginfo.sh
```

5. 然后可以使用 **scp**来获取数据。

其他资源

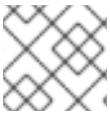
- 请参阅 在没有 SSH 的情况下在 OpenShift4 节点中生成 SOSREPORT 。

1.1.19. 后续步骤

- 自定义集群。
- 如果需要，您可以选择不使用远程健康报告。

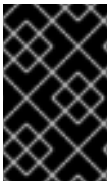
1.2. 在受限网络中在 IBM Z 和 LINUXONE 上安装集群

在 OpenShift Container Platform 版本 4.6 中，可以在受限网络中置备的 IBM Z 和 LinuxONE 基础架构上安装集群。



注意

虽然本文档仅涉及了 IBM Z，但它的所有信息也适用于 LinuxONE。



重要

非裸机平台还有其他注意事项。在尝试在此类环境中安装 OpenShift Container Platform 集群前，请参阅有关在未经测试的平台上部署 OpenShift Container Platform 的指南 中的信息。

先决条件

- 在受限网络中创建镜像 registry，并获取您的 OpenShift Container Platform 版本的 **imageContentSources** 数据。
- 在开始安装前，必须移动或删除现有的安装文件。这可保证在安装过程中创建和更新所需的安装文件。

  重要

  确定从可访问安装介质的机器中执行安装步骤。

- 为集群置备使用 NFS 的持久性存储。若要部署私有镜像 registry，您的存储必须提供 **ReadWriteMany** 访问模式。
- 查看有关 OpenShift Container Platform 安装和更新 流程的详细信息。
- 如果使用防火墙并计划使用遥测（telemetry），您必须将防火墙配置为允许集群需要访问的站点。

  注意

  如果您要配置代理，请务必也要查看此站点列表。

1.2.1. 关于在受限网络中安装

在 OpenShift Container Platform 4.6 中，可以执行不需要有效的互联网连接来获取软件组件的安装。受限网络安装可使用安装程序置备的基础架构或用户置备的基础架构完成，具体取决于您要安装集群的云平台。

如果选择在云平台中执行受限网络安装，仍然需要访问其云 API。有些云功能，比如 Amazon Web Service 的 Route 53 DNS 和 IAM 服务，需要访问互联网。根据您的网络，在裸机硬件或 VMware vSphere 上安装时可能需要较少的互联网访问。

要完成受限网络安装，您必须创建一个 registry，镜像 OpenShift Container Platform registry 的内容并包含其安装介质。您可以在堡垒主机上创建此镜像，该主机可同时访问互联网和您的封闭网络，也可以使用满足您的限制条件的其他方法。

重要

由于用户置备安装配置的复杂性，在尝试使用用户置备的基础架构受限网络安装前，请考虑完成标准用户置备的基础架构安装。通过完成此测试安装，您可以更轻松地隔离和排查您在受限网络中安装时可能出现的问题。

1.2.1.1. 其他限制

受限网络中的集群还有以下额外限制：

- **ClusterVersion** 状态包含一个 **Unable to retrieve available updates** 错误。
- 默认情况下，您无法使用 Developer Catalog 的内容，因为您无法访问所需的镜像流标签。

1.2.2. OpenShift Container Platform 的互联网访问

在 OpenShift Container Platform 4.6 中，您需要访问互联网来获得用来安装集群的镜像。

您必须具有以下互联网访问权限：

- 访问 OpenShift Cluster Manager 以下载安装程序并执行订阅管理。如果集群可以访问互联网，并且没有禁用 Telemetry，该服务会自动授权您的集群。
- 访问 Quay.io，以获取安装集群所需的软件包。
- 获取执行集群更新所需的软件包。

重要

如果您的集群无法直接访问互联网，则可以在置备的某些类基础架构上执行受限网络安装。在此过程中，您要下载所需的内容，并使用它在镜像 registry（mirror registry） 中填充安装集群并生成安装程序所需的软件包。对于某些安装类型，集群要安装到的环境不需要访问互联网。在更新集群之前，要更新 registry 镜像系统中的内容。

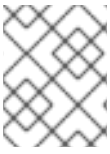
1.2.3. 具有用户置备基础架构的集群的机器要求

对于含有用户置备的基础架构的集群，您必须部署所有所需的机器。

1.2.3.1. 所需的机器

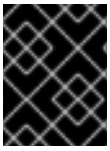
最小的 OpenShift Container Platform 集群需要下列主机：

- 一个临时 bootstrap 机器
- 三台 control plane 或 master 机器
- 至少两台计算机器，也称为 worker 机器。



注意

集群要求 bootstrap 机器在三台 control plane 机器上部署 OpenShift Container Platform 集群。您可在安装集群后删除 bootstrap 机器。



重要

要提高集群的高可用性，请在最少两个物理集群中的不同的 z/VM 实例中分布运行 control plane 机器。

bootstrap 和 control plane 机器必须使用 Red Hat Enterprise Linux CoreOS (RHCOS) 作为操作系统。但是，计算机器可以在 Red Hat Enterprise Linux CoreOS(RHCOS)或 Red Hat Enterprise Linux(RHEL)7.9 间进行选择。

请注意，RHCOS 基于 Red Hat Enterprise Linux（RHEL） 8，并继承其所有硬件认证和要求。请查看Red Hat Enterprise Linux 技术功能及限制 。

1.2.3.2. 网络连接要求

所有 Red Hat Enterprise Linux CoreOS (RHCOS) 机器在启动过程中需要 **initramfs** 中的网络从 Machine Config Server 获取 Ignition 配置文件。机器被配置为使用静态 IP 地址。不需要 DHCP 服务器。另外，集群中的每个 OpenShift Container Platform 节点都必须有权访问网络时间协议（NTP）服务器。

1.2.3.3. IBM Z 网络连接要求

要在 z/VM 中安装 IBM Z，您需要使用第 2 层模式的单一 z/VM 虚拟 NIC。您还需要：

- 直接连接的 OSA 或 RoCE 网络适配器
- z/VM vSwitch 设置。对于首选的设置，请使用 OSA 链接聚合。

1.2.3.4. 最低资源要求

每台集群机器都必须满足以下最低要求：

机器	操作系统	vCPU [1]	虚拟内存	存储	IOPS
bootstrap	RHCOS	4	16 GB	100 GB	N/A
Control plane	RHCOS	4	16 GB	100 GB	N/A
Compute	RHCOS	2	8 GB	100 GB	N/A

1. 当未启用并发多线程 (SMT) 或超线程时，一个 vCPU 相当于一个物理内核。启用后，使用以下公式来计算对应的比例：（每个内核数的线程）× sockets = vCPU。

1.2.3.5. 最低 IBM Z 系统环境

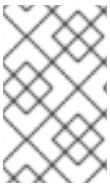
您可以在以下 IBM 硬件上安装 OpenShift Container Platform 版本 4.6：

- IBM z15（所有型号）、IBM z14（所有型号）、IBM z13 和 IBM z13s
- 任何版本的 LinuxONE

硬件要求

- 相当于为每个集群启用了 SMT2 的 IFL。

- 至少有一个网络连接连接到 **LoadBalancer** 服务，并为集群外的流量提供服务。



注意

您可以使用专用或共享 IFL 来分配充足的计算资源。资源共享是 IBM Z 的主要优势之一。但是，您必须在每个虚拟机监控程序层上正确调整容量，并确保每个 OpenShift Container Platform 集群有足够的资源。

重要

由于集群的整体性能可能会受到影响，因此用于设置 OpenShift Container Platform 集群的 LPAR 必须提供足够的计算能力。在这种情况下，虚拟机监控程序级别的 LPAR 权重管理、授权和 CPU 共享扮演重要角色。

操作系统要求

- 一个 z/VM 7.1 或更高版本的实例

在您的 z/VM 实例中设置：

- 3 个客户虚拟机作为 OpenShift Container Platform control plane 的机器
- 2 个客户虚拟机作为 OpenShift Container Platform 的计算机器
- 1 个客户虚拟机作为临时 OpenShift Container Platform bootstrap 机器

IBM Z 网络连接要求
要在 z/VM 中安装 IBM Z，您需要使用第 2 层模式的单一 z/VM 虚拟 NIC。您还需要：

- 直接连接的 OSA 或 RoCE 网络适配器
- z/VM vSwitch 设置。对于首选的设置，请使用 OSA 链接聚合。

z/VM 客户虚拟机的磁盘存储

- 附加了 FICON 的磁盘存储。可以是 z/VM Minidisks、fullpack Minidisks 或专用 DASD，它们都必须被格式化为 CDL，这是默认的 CDL。要达到 Red Hat Enterprise Linux CoreOS (RHCOS) 安装所需的最小 DASD 大小，您需要扩展地址卷 (EAV)。如果可用，使用 HyperPAV 来确保最佳性能。
- FCP 连接的磁盘存储

存储/主内存

- OpenShift Container Platform control plane 需要 16 GB
- OpenShift Container Platform 计算机器需要 8 GB
- 临时 OpenShift Container Platform bootstrap 机器需要 16 GB

1.2.3.6. 首选 IBM Z 系统环境

硬件要求

- 3 个与 6 个 IFL 等效的 LPARS，每个集群都启用了 SMT2。

- 用于连接 **LoadBalancer** 服务的两个网络连接，并为集群外的流量提供服务。
- HiperSockets，可直接作为设备附加到节点，或者与一个 z/VM VSWITCH 桥接以对 z/VM 客户机进行透明。要将 HiperSockets 直接连接到节点，您必须通过 RHEL 8 虚拟机设置到外部网络的网关来桥接到 HiperSockets 网络。

操作系统要求

- 2 个或 3 个 z/VM 7.1 或更高版本的实例以实现高可用性

在您的 z/VM 实例中设置：

- 3 个用于 OpenShift Container Platform control plane 机器的虚拟机，每个 z/VM 实例一个。
- 至少 6 个用于 OpenShift Container Platform 计算机器的虚拟机，分布在 z/VM 实例中。
- 1 个客户虚拟机作为临时 OpenShift Container Platform bootstrap 机器。
- 要确保在过量使用的环境中整合组件可用，请使用 CP 命令 **SET SHARE** 来增加 control plane 的优先级。对基础架构节点执行相同操作（如果存在）。请参阅 IBM 文档中的 SET SHARE。

IBM Z 网络连接要求
要在 z/VM 中安装 IBM Z，您需要使用第 2 层模式的单一 z/VM 虚拟 NIC。您还需要：

- 直接连接的 OSA 或 RoCE 网络适配器
- z/VM vSwitch 设置。对于首选的设置，请使用 OSA 链接聚合。

z/VM 客户虚拟机的磁盘存储

- 附加了 FICON 的磁盘存储。可以是 z/VM Minidisks、fullpack Minidisks 或专用 DASD，它们都必须被格式化为 CDL，这是默认的 CDL。要达到 Red Hat Enterprise Linux CoreOS (RHCOS) 安装所需的最小 DASD 大小，您需要扩展地址卷 (EAV)。如果可用，请使用 HyperPAV 和 High Performance FICON（zHPF）来确保最佳性能。
- FCP 连接的磁盘存储

存储/主内存

- OpenShift Container Platform control plane 需要 16 GB
- OpenShift Container Platform 计算机器需要 8 GB
- 临时 OpenShift Container Platform bootstrap 机器需要 16 GB

1.2.3.7. 证书签名请求管理

在使用您置备的基础架构时，集群只能有限地访问自动机器管理，因此您必须提供一种在安装后批准集群证书签名请求 (CSR) 的机制。**kube-controller-manager** 只能批准 kubelet 客户端 CSR。 **machine-approver** 无法保证使用 kubelet 凭证请求的提供证书的有效性，因为它不能确认是正确的机器发出了该请求。您必须决定并实施一种方法，以验证 kubelet 提供证书请求的有效性并进行批准。

其他资源

- 请参阅 IBM 文档中的使用 z/VM 虚拟交换机桥接 HiperSockets LAN。
- 请参阅在 z/VM 的 Linux 客户端中扩展 HyperPAV 别名设备 以获得性能优化。

- 有关 LPAR 权重管理和权利，请参阅 LPAR 性能中的主题。

1.2.4. 创建用户置备的基础架构

在部署采用用户置备的基础架构的 OpenShift Container Platform 集群前，您必须创建底层基础架构。

先决条件

- 在为集群创建支持基础架构之前，请参阅OpenShift Container Platform 4.x Tested Integrations页。

流程

1. 在每个节点上配置 DHCP 或设置静态 IP 地址。
2. 提供所需的负载均衡器。
3. 配置机器的端口。
4. 配置 DNS。
5. 确保网络可以正常工作。

1.2.4.1. 用户置备的基础架构对网络的要求

所有 Red Hat Enterprise Linux CoreOS（RHCOS）机器在启动过程中需要 **initramfs** 中的网络从机器配置服务器获取 Ignition 配置。

在初次启动过程中，需要一个 DHCP 服务器或集群中的每个机器都设置了静态 IP 地址来建立网络连接，以下载它们的 Ignition 配置文件。

建议您使用 DHCP 服务器为集群进行长期机器管理。确保 DHCP 服务器已配置为向集群机器提供持久 IP 地址和主机名。

Kubernetes API 服务器必须能够解析集群机器的节点名称。如果 API 服务器和 worker 节点位于不同的区域中，您可以配置默认 DNS 搜索区域，以便 API 服务器能够解析节点名称。另一种支持的方法是始终在节点对象和所有 DNS 请求中使用完全限定域名来指代主机。

您必须配置机器间的网络连接，以便集群组件进行通信。每台机器都必须能够解析集群中所有其他机器的主机名。

表 1.10. 所有机器到所有机器

协议	端口	描述
ICMP	N/A	网络可访问性测试
TCP	**1936**	指标
	9000-**9999**	主机级别的服务，包括端口 **9100**-**9101** 上的节点导出器和端口 **9099** 上的 Cluster Version Operator。
	10250-**10259**	Kubernetes 保留的默认端口

协议	端口	描述
	10256	openshift-sdn
UDP	**4789**	VXLAN 和 Geneve
	6081	VXLAN 和 Geneve
	9000-9999	主机级别的服务，包括端口 **9100-9101** 上的节点导出器。
TCP/UDP	**30000-32767**	Kubernetes 节点端口

表 1.11. 要通过控制平面的所有机器

协议	端口	描述
TCP	**6443**	Kubernetes API

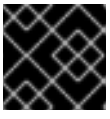
表 1.12. control plane 机器到 control plane 机器

协议	端口	描述
TCP	**2379-2380**	etcd 服务器和对等端口

网络拓扑要求
您为集群置备的基础架构必须满足下列网络拓扑要求。

负载均衡器
在安装 OpenShift Container Platform 前，您必须置备两个满足以下要求的负载均衡器：

1. **API 负载均衡器**：提供一个通用端点，供用户（包括人和机器）与平台交互和配置。配置以下条件：
   - 只适用于第 4 层负载均衡。这可被称为 Raw TCP、SSL Passthrough 或者 SSL 桥接模式。如果使用 SSL Bridge 模式，必须为 API 路由启用 Server Name Indication（SNI）。
   - 无状态负载平衡算法。这些选项根据负载均衡器的实现而有所不同。



重要

不要为 API 负载均衡器配置会话持久性。

在负载均衡器的前端和后台配置以下端口：

表 1.13. API 负载均衡器

端口	后端机器（池成员）	内部	外部	描述
6443	Bootstrap 和 control plane.bootstrap 机器初始化集群 control plane 后，您要从负载均衡器中删除 bootstrap 机器。您必须为 API 服务器健康检查探测配置 **/readyz** 端点。	X	X	Kubernetes API 服务器
22623	Bootstrap 和 control plane.bootstrap 机器初始化集群 control plane 后，您要从负载均衡器中删除 bootstrap 机器。	X		机器配置服务器

注意

负载均衡器必须配置为，从 API 服务器关闭 **/readyz** 端点到从池中删除 API 服务器实例时最多需要 30 秒。在 **/readyz** 返回错误或处于健康状态后的时间范围内，端点必须被删除或添加。每 5 秒或 10 秒探测一次，有两个成功请求处于健康状态，三个成为不健康的请求经过测试。

2. **应用程序入口负载均衡器**:提供来自集群外部的应用程序流量流量的 Ingress 点。配置以下条件：

   - 只适用于第 4 层负载均衡。这可被称为 Raw TCP、SSL Passthrough 或者 SSL 桥接模式。如果使用 SSL Bridge 模式，您必须为 Ingress 路由启用Server Name Indication（SNI）。
   - 建议根据可用选项以及平台上托管的应用程序类型，使用基于连接的或者基于会话的持久性。

   在负载均衡器的前端和后台配置以下端口：

表 1.14. 应用程序入口负载均衡器

端口	后端机器（池成员）	内部	外部	描述
443	默认运行入口路由器 Pod、计算或 worker 的机器。	X	X	HTTPS 流量
80	默认运行入口路由器 Pod、计算或 worker 的机器。	X	X	HTTP 流量

提示

如果负载均衡器可以看到客户端的真实 IP 地址，启用基于 IP 的会话持久性可提高使用端到端 TLS 加密的应用程序的性能。

注意

OpenShift Container Platform 集群需要正确配置入口路由器。control plane 初始化后，您必须配置入口路由器。

NTP 配置

OpenShift Container Platform 集群默认配置为使用公共网络时间协议（NTP）服务器。如果要使用本地企业 NTP 服务器，或者集群部署在断开连接的网络中，您可以将集群配置为使用特定的时间服务器。如需更多信息，请参阅*配置 chrony 时间服务*的文档。

如果 DHCP 服务器提供 NTP 服务器信息，Red Hat Enterprise Linux CoreOS（RHCOS）机器上的 chrony 时间服务会读取信息，并可与 NTP 服务器同步时钟。:!restricted:

其他资源

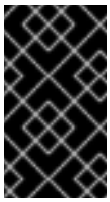
- 配置 chrony 时间服务

1.2.4.2. 用户置备 DNS 要求

DNS 用于名称解析和反向名称解析。DNS A/AAAA 或 CNAME 记录用于名称解析，PTR 记录用于反向解析名称。反向记录很重要，因为 Red Hat Enterprise Linux CoreOS（RHCOS）使用反向记录为所有节点设置主机名。另外，反向记录用于生成 OpenShift Container Platform 需要操作的证书签名请求（CSR）。

采用用户置备的基础架构的 OpenShift Container Platform 集群需要以下 DNS 记录。在每一记录中，**<cluster_name>** 是集群名称，**<base_domain>** 则是您在 **install-config.yaml** 文件中指定的集群基域。完整的 DNS 记录采用如下格式: **<component>.<cluster_name>.<base_domain>.**。

表 1.15. 所需的 DNS 记录

组件	记录	描述
Kubernetes API	**api.<cluster_name>.<base_domain>.**	添加 DNS A/AAAA 或 CNAME 记录，以及 DNS PTR 记录，以识别 control plane 机器的负载均衡器。这些记录必须由集群外的客户端以及集群中的所有节点解析。
	api-int.<cluster_name>.<base_domain>.	添加 DNS A/AAAA 或 CNAME 记录，以及 DNS PTR 记录，以识别 control plane 机器的负载均衡器。这些记录必须可以从集群中的所有节点解析。  **重要** API 服务器必须能够根据在 Kubernetes 中记录的主机名解析 worker 节点。如果 API 服务器无法解析节点名称，则代理的 API 调用会失败，且您无法从 pod 检索日志。
Routes	***.apps.<cluster_name>.<base_domain>.**	添加通配符 DNS A/AAAA 或 CNAME 记录，指向以运行入口路由器 Pod 的机器（默认为 worker 节点）为目标的负载均衡器。这些记录必须由集群外的客户端以及集群中的所有节点解析。
bootstrap	**bootstrap.<cluster_name>.<base_domain>.**	添加 DNS A/AAAA 或 CNAME 记录，以及 DNS PTR 记录来识别 bootstrap 机器。这些记录必须由集群中的节点解析。
Master 主机	**<master><n>.<cluster_name>.<base_domain>.**	DNS A/AAAA 或 CNAME 记录，以识别 control plane 节点（也称为 master 节点）的每台机器。这些记录必须由集群中的节点解析。

组件	记录	描述
Worker 主机	**<worker><n>. <cluster_name>. <base_domain>.**	添加 DNS A/AAAA 或 CNAME 记录，以识别 worker 节点的每台机器。这些记录必须由集群中的节点解析。

提示

您可以使用 **nslookup <hostname>** 命令来验证名称解析。您可以使用 **dig -x <ip_address>** 命令来验证 PTR 记录的反向名称解析。

下面的 BIND 区文件的例子展示了关于名字解析的 A 记录的例子。这个示例的目的是显示所需的记录。这个示例不是为选择一个名称解析服务提供建议。

例 1.3. DNS 区数据库示例

```
$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 IN A 192.168.1.5
smtp IN A 192.168.1.5
;
helper IN A 192.168.1.5
helper.ocp4 IN A 192.168.1.5
;
; The api identifies the IP of your load balancer.
api.ocp4  IN A 192.168.1.5
api-int.ocp4  IN A 192.168.1.5
;
; The wildcard also identifies the load balancer.
*.apps.ocp4  IN A 192.168.1.5
;
; Create an entry for the bootstrap host.
bootstrap.ocp4 IN A 192.168.1.96
;
; Create entries for the master hosts.
master0.ocp4  IN A 192.168.1.97
master1.ocp4  IN A 192.168.1.98
master2.ocp4  IN A 192.168.1.99
;
; Create entries for the worker hosts.
worker0.ocp4  IN A 192.168.1.11
worker1.ocp4  IN A 192.168.1.7
;
;EOF
```

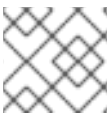
下面的 BIND 区文件示例显示了反向名字解析的 PTR 记录示例。

例 1.4. 反向记录的 DNS 区数据库示例

```
$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.
;
; The syntax is "last octet" and the host must have an FQDN
; with a trailing dot.
97 IN PTR master0.ocp4.example.com.
98 IN PTR master1.ocp4.example.com.
99 IN PTR master2.ocp4.example.com.
;
96 IN PTR bootstrap.ocp4.example.com.
;
5 IN PTR api.ocp4.example.com.
5 IN PTR api-int.ocp4.example.com.
;
11 IN PTR worker0.ocp4.example.com.
7 IN PTR worker1.ocp4.example.com.
;
;EOF
```

1.2.5. 生成 SSH 私钥并将其添加到代理中

如果要在集群上执行安装调试或灾难恢复，则必须为 **ssh-agent** 和安装程序提供 SSH 密钥。您可以使用此密钥访问公共集群中的 bootstrap 机器来排除安装问题。



注意

在生产环境中，您需要进行灾难恢复和调试。

重要

不要在生产环境中跳过这个过程，因为生产环境需要灾难恢复和调试。

您可以使用此密钥以 **core** 用户身份通过 SSH 连接到 master 节点。在部署集群时，此密钥会添加到 **core** 用户的 **~/.ssh/authorized_keys** 列表中。

流程

1. 如果还没有为计算机上免密码身份验证而配置的 SSH 密钥，请创建一个。例如，在使用 Linux 操作系统的计算机上运行以下命令：

```
$ ssh-keygen -t ed25519 -N '' \
  -f <path>/<file_name> 1
```

1 指定新 SSH 密钥的路径和文件名，如 **~/.ssh/id_rsa**。如果您已有密钥对，请确保您的公钥位于 **~/.ssh** 目录中。

运行此命令会在指定的位置生成不需要密码的 SSH 密钥。

注意

如果您计划在 **x86_64** 架构中安装使用 FIPS 验证的/Modules in Process 加密库的 OpenShift Container Platform 集群，不要创建使用 **ed25519** 算法的密钥。反之，创建一个使用 **rsa** 或 **ecdsa** 算法的密钥。

2. 作为后台任务启动 **ssh-agent** 进程：

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

输出示例

```
Agent pid 31874
```

注意

如果您的集群采用 FIPS 模式，则只使用 FIPS 兼容算法来生成 SSH 密钥。密钥必须是 RSA 或 ECDSA。

3. 将 SSH 私钥添加到 **ssh-agent**：

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

输出示例

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

1 指定 SSH 私钥的路径和文件名，如 **~/.ssh/id_rsa**

后续步骤

- 在安装 OpenShift Container Platform 时，为安装程序提供 SSH 公钥。

1.2.6. 手动创建安装配置文件

对于使用用户置备的基础架构的 OpenShift Container Platform 安装，您必须手动生成安装配置文件。

先决条件

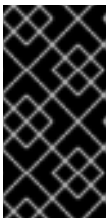
- 获取 OpenShift Container Platform 安装程序和集群的访问令牌。

- 获取命令输出中的 **imageContentSources** 部分来镜像存储库。
- 获取您的镜像 registry 的证书内容。

流程

1. 创建用来存储您所需的安装资产的安装目录：

```
$ mkdir <installation_directory>
```



重要

您必须创建目录。一些安装信息，如 bootstrap X.509 证书，有较短的过期间隔，因此不要重复使用安装目录。如果要重复使用另一个集群安装中的个别文件，可以将其复制到您的目录中。但是，一些安装数据的文件名可能会在发行版本之间有所改变。从 OpenShift Container Platform 老版本中复制安装文件时要格外小心。

2. 自定义以下 **install-config.yaml** 文件模板，并将它保存到 **<installation_directory>** 中。

注意

此配置文件必须命名为 **install-config.yaml**。

- 除非使用 RHCOS 默认信任的 registry，如 **docker.io**，否则必须在 **additionalTrustBundle** 部分中提供镜像存储库的证书内容。在大多数情况下，必须为您的镜像提供证书。
- 您必须包含命令输出中的 **imageContentSources** 部分，才能镜像存储库。

3. 备份 **install-config.yaml** 文件，以便用于安装多个集群。

重要

install-config.yaml 文件会在安装过程的下一步骤中消耗掉。现在必须备份它。

1.2.6.1. 安装配置参数

在部署 OpenShift Container Platform 集群前，您可以提供参数值，以描述托管集群的云平台的帐户并选择性地自定义集群平台。在创建 **install-config.yaml** 安装配置文件时，您可以通过命令行来提供所需的参数的值。如果要自定义集群，可以修改 **install-config.yaml** 文件来提供关于平台的更多信息。

注意

安装之后，您无法修改 **install-config.yaml** 文件中的这些参数。

重要

openshift-install 命令不验证参数的字段名称。如果指定了不正确的名称，则不会创建相关的文件或对象，且不会报告错误。确保所有指定的参数的字段名称都正确。

1.2.6.1.1. 所需的配置参数

下表描述了所需的安装配置参数：

表 1.16. 所需的参数

参数	描述	值
apiVersion	**install-config.yaml** 内容的 API 版本。当前版本是 **v1**。安装程序还可能支持旧的 API 版本。	字符串
baseDomain	云供应商的基域。此基础域用于创建到 OpenShift Container Platform 集群组件的路由。集群的完整 DNS 名称是 **baseDomain** 和 **metadata.name** 参数值的组合，其格式为 **<metadata.name>.<baseDomain>**。	完全限定域名或子域名，如 **example.com**。
metadata	Kubernetes 资源 **ObjectMeta**，其中只消耗 **name** 参数。	对象
metadata.name	集群的名称。集群的 DNS 记录是 **{{.metadata.name}}.{{.baseDomain}}** 的子域。	小写字母,连字符(-)和句点(.)的字符串，如 **dev**。
platform	执行安装的具体平台配置：**aws**、**baremetal**、**azure**、**openstack**、**ovirt**、**vsphere**。有关 **platform.<platform>** 参数的额外信息，请参考下表来了解您的具体平台。	对象

pullSecret	从 Red Hat OpenShift Cluster Manager 获取 pull secret，验证从 Quay.io 等服务中下载 OpenShift Container Platform 组件的容器镜像。	{
		"auths":{
		"cloud.openshift.com":{
		"auth":"b3Blb=",
		"email":"you@example.com"
		},
		"quay.io":{
		"auth":"b3Blb=",
		"email":"you@example.com"
		}
		}
		}

1.2.6.1.2. 网络配置参数

您可以根据现有网络基础架构的要求自定义安装配置。例如，您可以扩展集群网络的 IP 地址块，或者提供不同于默认值的不同 IP 地址块。

只支持 IPv4 地址。

表 1.17. 网络参数

参数	描述	值
networking	集群网络的配置。	对象  **注意** 您不能在安装后修改 **networking** 对象指定的参数。
networking.networkType	要安装的集群网络供应商 Container Network Interface（CNI）插件。	**OpenShiftSDN** 或 **OVNKubernetes**。默认值为 **OpenShiftSDN**。
networking.clusterNetwork	pod 的 IP 地址块。 默认值为 **10.128.0.0/14**，主机前缀为 **/23**。 如果您指定多个 IP 地址块，则块不得互相重叠。	一个对象数组。例如： networking: clusterNetwork: - cidr: 10.128.0.0/14 hostPrefix: 23
networking.clusterNetwork.cidr	使用 **networking.clusterNetwork** 时需要此项。IP 地址块。 一个 IPv4 网络。	使用 CIDR 形式的 IP 地址块。IPv4 块的前缀长度介于 **0** 到 **32** 之间。
networking.clusterNetwork.hostPrefix	分配给每个单独节点的子网前缀长度。例如，如果 **hostPrefix** 设为 **23**，则每个节点从所给的 **cidr** 中分配一个 **/23** 子网。**hostPrefix** 值 **23** 提供 510（2^(32 - 23)- 2）个 pod IP 地址。	子网前缀。 默认值为 **23**。
networking.serviceNetwork	服务的 IP 地址块。默认值为 **172.30.0.0/16**。 OpenShift SDN 和 OVN-Kubernetes 网络供应商只支持服务网络的一个 IP 地址块。	CIDR 格式具有 IP 地址块的数组。例如： networking: serviceNetwork: - 172.30.0.0/16

参数	描述	值
networking.machineNetwork	机器的 IP 地址块。 如果您指定多个 IP 地址块，则块不得互相重叠。 如果您指定了多个 IP 内核参数，**machineNetwork.cidr** 值必须是主网络的 CIDR。	一个对象数组。例如： networking: machineNetwork: - cidr: 10.0.0.0/16
networking.machineNetwork.cidr	使用 **networking.machineNetwork** 时需要。IP 地址块。libvirt 以外的所有平台的默认值为 **10.0.0.0/16**。对于 libvirt，默认值为 **192.168.126.0/24**。	CIDR 表示法中的 IP 网络块。 例如：**10.0.0.0/16**。 **注意** 将 **networking.machineNetwork** 设置为与首选 NIC 所在的 CIDR 匹配。

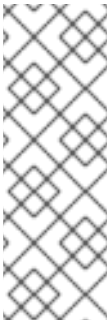
1.2.6.1.3. 可选配置参数

下表描述了可选安装配置参数：

表 1.18. 可选参数

参数	描述	值
additionalTrustBundle	添加到节点可信证书存储中的 PEM 编码 X.509 证书捆绑包。配置了代理时，也可以使用这个信任捆绑包。	字符串
compute	组成计算节点的机器的配置。	machine-pool 对象的数组。详情请查看以下"Machine-pool"表。
compute.architecture	决定池中机器的指令集合架构。目前不支持异构集群，因此所有池都必须指定相同的架构。有效值为 **amd64** （默认值）。	字符串

参数	描述	值
compute.hyperthreading	是否在计算机器上启用或禁用并发多线程或**超线程**。默认情况下，启用并发多线程以提高机器内核的性能。  **重要** 如果禁用并发多线程，请确保在容量规划时考虑到机器性能可能会显著降低的问题。	**Enabled** 或 **Disabled**
compute.name	使用 **compute** 时需要此值。机器池的名称。	**worker**
compute.platform	使用 **compute** 时需要此值。使用此参数指定托管 worker 机器的云供应商。此参数值必须与 **controlPlane.platform** 参数值匹配。	**aws**、**azure**、**gcp**、**openstack**、**ovirt**、**vsphere** 或 **{}**
compute.replicas	要置备的计算机器数量，也称为 worker 机器。	大于或等于 **2** 的正整数。默认值为 **3**。
controlPlane	组成 control plane 的机器的配置。	**MachinePool** 对象的数组。详情请查看以下"Machine-pool"表。
controlPlane.architecture	决定池中机器的指令集合架构。目前不支持异构集群，因此所有池都必须指定相同的架构。有效值为 **amd64** （默认值）。	字符串
controlPlane.hyperthreading	是否在 control plane 机器上启用或禁用并发多线程或**超线程**。默认情况下，启用并发多线程以提高机器内核的性能。  **重要** 如果禁用并发多线程，请确保在容量规划时考虑到机器性能可能会显著降低的问题。	**Enabled** 或 **Disabled**
controlPlane.name	使用 **controlPlane** 时需要。机器池的名称。	**master**

参数	描述	值
controlPlane.platform	使用 **controlPlane** 时需要。使用此参数指定托管 control plane 机器的云供应商。此参数值必须与 **compute.platform** 参数值匹配。	**aws**、**azure**、**gcp**、**openstack**、**ovirt**、**vsphere** 或 **{}**
controlPlane.replicas	要置备的 control plane 机器数量。	唯一支持的值是 **3**，它是默认值。
credentialsMode	Cloud Credential Operator（CCO）模式。如果没有指定任何模式，CCO 会动态地尝试决定提供的凭证的功能，在支持多个模式的平台上使用 mint 模式。  **注意** 不是所有 CCO 模式都支持所有云供应商。如需有关 CCO 模式的更多信息，请参阅*Red Hat Operator 参考指南*内容中的 *Cloud Credential Operator* 条目。	**Mint**、**Passthrough**、**Manual** 或空字符串(**""**)。
fips	启用或禁用 FIPS 模式。默认为 **false**（禁用）。如果启用了 FIPS 模式，运行 OpenShift Container Platform 的 Red Hat Enterprise Linux CoreOS (RHCOS) 机器会绕过默认的 Kubernetes 加密套件，并使用由 RHCOS 提供的加密模块。  **重要** 只有在 **x86_64** 架构中的 OpenShift Container Platform 部署支持 FIPS 验证的/Modules in Process 加密库。  **注意** 如果使用 Azure File 存储，则无法启用 FIPS 模式。	**false** 或 **true**
imageContentSources	release-image 内容的源和仓库。	对象数组。包括一个 **source** 以及可选的 **mirrors**，如下表所示。

参数	描述	值
imageContentSources.source	使用 **imageContentSources** 时需要。指定用户在镜像拉取规格中引用的仓库。	字符串
imageContentSources.mirrors	指定可能还包含同一镜像的一个或多个仓库。	字符串数组
publish	如何发布或公开集群的面向用户的端点，如 Kubernetes API、OpenShift 路由。	**Internal** 或 **External**。默认值为 **External**。 在非云平台上不支持将此字段设置为 **Internal**。  **重要** 如果将字段的值设为 **Internal**，集群将无法运行。如需更多信息，请参阅 BZ#1953035。
sshKey	用于验证集群机器访问的 SSH 密钥或密钥。  **注意** 对于您要在其上执行安装调试或灾难恢复的生产环境 OpenShift Container Platform 集群，请指定 **ssh-agent** 进程使用的 SSH 密钥。	一个或多个密钥。例如： sshKey: <key1> <key2> <key3>

1.2.6.2. IBM Z 的 install-config.yaml 文件示例

您可以自定义 **install-config.yaml** 文件，以指定有关 OpenShift Container Platform 集群平台的更多信息，或修改所需参数的值。

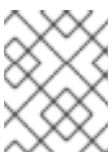
```
apiVersion: v1
baseDomain: example.com (1)
compute: (2)
- hyperthreading: Enabled (3)
  name: worker
```

```
  replicas: 0 4
  architecture : s390x
controlPlane: 5
  hyperthreading: Enabled 6
  name: master
  replicas: 3 7
  architecture : s390x
metadata:
  name: test 8
networking:
  clusterNetwork:
  - cidr: 10.128.0.0/14 9
    hostPrefix: 23 10
  networkType: OpenShiftSDN
  serviceNetwork: 11
  - 172.30.0.0/16
platform:
  none: {} 12
fips: false 13
pullSecret: '{"auths":{"<local_registry>": {"auth": "<credentials>","email": "you@example.com"}}}' 14
sshKey: 'ssh-ed25519 AAAA...' 15
additionalTrustBundle: | 16
  -----BEGIN CERTIFICATE-----
  ZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZZ
  -----END CERTIFICATE-----
imageContentSources: 17
- mirrors:
  - <local_repository>/ocp4/openshift4
  source: quay.io/openshift-release-dev/ocp-release
- mirrors:
  - <local_repository>/ocp4/openshift4
  source: quay.io/openshift-release-dev/ocp-v4.0-art-dev
```

1 集群的基域。所有 DNS 记录都必须是这个基域的子域，并包含集群名称。

2 5 **controlPlane** 部分是一个单映射，但 **compute** 部分是一系列映射。为满足不同数据结构的要求，**compute** 部分的第一行必须以连字符 **-** 开头，**controlPlane** 部分的第一行则不可以连字符开头。只使用一个 control plane 池。

3 6 是否要启用或禁用并发多线程（SMT）或**超线程**。默认情况下，启用 SMT 可提高机器内核的性能。您可以通过将参数值设为 **Disabled** 来禁用。如果禁用 SMT，则必须在所有集群机器中禁用它，其中包括 control plane 和计算机器。



注意

默认启用并发多线程（SMT）。如果在 BIOS 设置中没有启用 SMT，**hyperthreading** 参数不会起作用。

重要

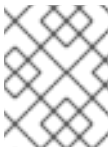
如果您禁用 **hyperthreading**（无论是在 BIOS 中还是在 **install-config.yaml** 中），请确保您对可能会造成的机器性能显著降低的情况有所考虑。

4 **replicas** 参数的值必须设置为 **0**。此参数控制集群为您创建和管理的 worker 数量，使用用户置备的基础架构时集群不会执行这些功能。在完成 OpenShift Container Platform 安装前，您必须手动为集

7 您添加到集群的 control plane 机器数量。由于集群将这个值用作集群中 etcd 端点的数量，因此该值必须与您部署的 control plane 机器数量匹配。

8 您在 DNS 记录中指定的集群名称。

9 从中分配 pod IP 地址的 IP 地址块。此块不得与现有的物理网络重叠。这些 IP 地址用于 pod 网络。如果您需要从外部网络访问 pod，请配置负载均衡器和路由器来管理流量。



注意

类 E CIDR 范围保留给以后使用。要使用 Class E CIDR 范围，您必须确保您的网络环境接受 Class E CIDR 范围内的 IP 地址。

10 分配给每个单独节点的子网前缀长度。例如，如果 **hostPrefix** 设为 **23**，则每个节点从所给的 **cidr** 中分配一个 **/23** 子网，这样就能有 510 (2^(32 - 23) - 2) 个 Pod IP 地址。如果您需要从外部网络访问节点，请配置负载均衡器和路由器来管理流量。

11 用于服务 IP 地址的 IP 地址池。您只能输入一个 IP 地址池。此块不得与现有的物理网络重叠。如果您需要从外部网络访问服务，请配置负载均衡器和路由器来管理流量。

12 您必须将平台设置为 **none**。您无法为 IBM Z 基础架构提供额外的平台配置变量。

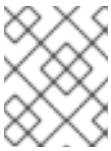
13 是否启用或禁用 FIPS 模式。默认情况下不启用 FIPS 模式。如果启用了 FIPS 模式，运行 OpenShift Container Platform 的 Red Hat Enterprise Linux CoreOS (RHCOS) 机器会绕过默认的 Kubernetes 加密套件，并使用由 RHCOS 提供的加密模块。

重要

只有在 **x86_64** 架构中的 OpenShift Container Platform 部署支持 FIPS 验证的/Modules in Process 加密库。

14 对于 **<local_registry>**，请指定 registry 域名，以及您的镜像 registry 用来提供内容的可选端口。例如： **registry.example.com** 或者 **registry.example.com:5000**。使用 **<credentials>** 为您生成的镜像 registry 指定 base64 编码的用户名和密码。

15 Red Hat Enterprise Linux CoreOS (RHCOS) 中 **core** 用户的默认 SSH 密钥的公钥部分。



注意

对于您要在其上执行安装调试或灾难恢复的生产环境 OpenShift Container Platform 集群，请指定 **ssh-agent** 进程使用的 SSH 密钥。

16 添加 **additionalTrustBundle** 参数和值。该值必须是您用于镜像 registry 的证书文件内容，可以是现有的可信证书颁发机构或您为镜像 registry 生成的自签名证书。

17 提供命令输出中的 **imageContentSources** 部分来镜像存储库。

1.2.6.3. 在安装过程中配置集群范围代理

生产环境可能会拒绝直接访问互联网，而是提供 HTTP 或 HTTPS 代理。您可以通过在 **install-config.yaml** 文件中配置代理设置，将新的 OpenShift Container Platform 集群配置为使用代理。

先决条件

- 您有一个现有的 **install-config.yaml** 文件。
- 您检查了集群需要访问的站点，并决定是否需要绕过代理。默认情况下代理所有集群出口流量，包括对托管云供应商 API 的调用。您需要将站点添加到 **Proxy** 对象的 **spec.noProxy** 字段来绕过代理。

注意

Proxy 对象 **status.noProxy** 字段使用安装配置中的 **networking.machineNetwork[].cidr**、**networking.clusterNetwork[].cidr** 和 **networking.serviceNetwork[]** 字段的值填充。

对于在 Amazon Web Services(AWS)、Google Cloud Platform(GCP)、Microsoft Azure 和 Red Hat OpenStack Platform(RHOSP)上安装, **Proxy** 对象 **status.noProxy** 字段也会使用实例元数据端点填充(**169.254.169.254**)。

流程

1. 编辑 **install-config.yaml** 文件并添加代理设置。例如：

```
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-----
...
```

1. 用于创建集群外 HTTP 连接的代理 URL。URL 必须是 **http**。
2. 用于创建集群外 HTTPS 连接的代理 URL。
3. 要排除在代理中的目标域名、IP 地址或其他网络 CIDR 的逗号分隔列表。在域前面加 **.** 来仅匹配子域。例如： **.y.com** 匹配 **x.y.com**，但不匹配 **y.com**。使用 ***** 绕过所有目的地的代理。
4. 如果提供，安装程序会在 **openshift-config** 命名空间中生成名为 **user-ca-bundle** 的配置映射，以容纳额外的 CA 证书。如果您提供 **additionalTrustBundle** 和至少一个代理设置，**Proxy** 对象会被配置为引用 **trustedCA** 字段中的 **user-ca-bundle** 配置映射。然后，Cluster Network Operator 会创建一个 **trusted-ca-bundle** 配置映射，将 **trustedCA** 参数指定的值与 RHCOS 信任捆绑包合并。**additionalTrustBundle** 字段是必需的，除非代理的身份证书由来自 RHCOS 信任捆绑包的颁发机构签名。

注意

安装程序不支持代理的 **readinessEndpoints** 字段。

2. 保存该文件，并在安装 OpenShift Container Platform 时引用。

安装程序会创建一个名为 **cluster** 的集群范围代理，该代理使用提供的 **install-config.yaml** 文件中的代理设置。如果没有提供代理设置，仍然会创建一个 **cluster Proxy** 对象，但它会有一个空 **spec**。

注意

只支持名为 **cluster** 的 **Proxy** 对象，且无法创建额外的代理。

1.2.7. 创建 Kubernetes 清单和 Ignition 配置文件

由于您必须修改一些集群定义文件并要手动启动集群机器，因此您必须生成 Kubernetes 清单和 Ignition 配置文件，集群需要这两项来创建其机器。

安装配置文件转换为 Kubernetes 清单。清单嵌套到 Ignition 配置文件中，稍后用于创建集群。

重要

- 安装程序生成的 Ignition 配置文件包含在 24 小时后过期的证书，然后在过期时进行续订。如果在更新证书前关闭集群，且集群在 24 小时后重启，集群会自动恢复过期的证书。一个例外情况是，您需要手动批准待处理的 **node-bootstrapper** 证书签名请求（CSR）来恢复 kubelet 证书。如需更多信息，请参阅*从过期的 control plane 证书中恢复*的文档。
- 建议您在生成 12 小时后使用 Ignition 配置文件，因为集群安装后 24 小时证书从 16 小时轮转至 22 小时。通过在 12 小时内使用 Ignition 配置文件，您可以避免在安装过程中运行证书更新时避免安装失败。

先决条件

- 已获得 OpenShift Container Platform 安装程序。
- 已创建 **install-config.yaml** 安装配置文件。

流程

1. 切换到包含安装程序的目录，并为集群生成 Kubernetes 清单：

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

1 对于 **<installation_directory>**，请指定含有您创建的 **install-config.yaml** 文件的安装目录。

2. 检查 **<installation_directory>/manifests/cluster-scheduler-02-config.yml** Kubernetes 清单文件中的 **mastersSchedulable** 参数是否已设置为 **false**。此设置可防止在 control plane 机器上调度 pod：

   a. 打开 **<installation_directory>/manifests/cluster-scheduler-02-config.yml** 文件。

   b. 找到 **mastersSchedulable** 参数并确保它被设置为 **false**。

c. 保存并退出文件。

3. 要创建 Ignition 配置文件，从包含安装程序的目录运行以下命令：

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

1 对于 **<installation_directory>**，请指定相同的安装目录。

该目录中将生成以下文件：

```
.
├── auth
│   ├── kubeadmin-password
│   └── kubeconfig
├── bootstrap.ign
├── master.ign
├── metadata.json
└── worker.ign
```

1.2.8. 创建 Red Hat Enterprise Linux CoreOS (RHCOS) 机器

在您置备的 IBM Z 环境中安装集群前，您必须在 z/VM 虚拟机上安装 RHCOS 以便集群使用。完成以下步骤以创建机器。

先决条件

- 在置备机器中运行 FTP 服务器，您创建的机器需要可以访问这个 FTP 服务器。

流程

1. 在您置备的机器上登录到 Linux。
2. 从 RHCOS 镜像镜像 获取 Red Hat Enterprise Linux CoreOS（RHCOS）内核、initramfs 和 rootfs 文件。

重要

RHCOS 镜像可能不会随着 OpenShift Container Platform 的每一发行版本都有改变。您必须下载最高版本的镜像，其版本号应小于或等于您安装的 OpenShift Container Platform 版本。只使用以下流程中描述的适当内核、initramfs 和 rootfs 工件。

文件名包含 OpenShift Container Platform 版本号。它们类似以下示例：

- kernel: **rhcos-<version>-live-kernel-<architecture>**
- initramfs: **rhcos-<version>-live-initramfs.<architecture>.img**
- rootfs: **rhcos-<version>-live-rootfs.<architecture>.img**

注意

FCP 和 DASD 的 rootfs 镜像是相同的。

3. 创建参数文件。以下参数特定于特定虚拟机：

   - 对于 **coreos.inst.install_dev=**，请为 DASD 安装指定 **dasda**，或者为 FCP 指定 **sda**。请注意 FCP 需要 **zfcp.allow_lun_scan=0**。
   - 对于 **rd.dasd=**，请指定要安装 RHCOS 的 DASD。
   - **rd.zfcp=<adapter>,<wwpn>,<lun>** 指定要在其中安装 RHCOS 的 FCP 磁盘。
   - 对于 **ip=**，请指定以下七项：

     i. 机器的 IP 地址。

     ii. 一个空字符串。

     iii. 网关

     iv. 子网掩码。

     v. **hostname.domainname**格式的机器主机和域名。省略这个值会让 RHCOS 来决定这个值。

     vi. 网络接口名称。省略这个值会让 RHCOS 来决定这个值。

     vii. 如果使用静态 IP 地址，则为一个空字符串。

   - 对于 **coreos.inst.ignition_url=**，为机器角色指定 Ignition 文件。使用 **bootstrap.ign**、**master.ign** 或 **worker.ign**。只支持 HTTP 和 HTTPS 协议。
   - 对于 **coreos.live.rootfs_url=**，为您引导的内核和 initramfs 指定匹配的 rootfs 工件。只支持 HTTP 和 HTTPS 协议。
   - 所有其他参数都可以保留。
     bootstrap 机器的实例参数文件（**bootstrap-0.parm**）如下：

     ```
     rd.neednet=1 \
     console=ttysclp0 \
     coreos.inst.install_dev=dasda \
     coreos.live.rootfs_url=http://cl1.provide.example.com:8080/assets/rhcos-live-rootfs.s390x.img \
     coreos.inst.ignition_url=http://cl1.provide.example.com:8080/ignition/bootstrap.ign \
     ip=172.18.78.2::172.18.78.1:255.255.255.0:::none nameserver=172.18.78.1 \
     rd.znet=qeth,0.0.bdf0,0.0.bdf1,0.0.bdf2,layer2=1,portno=0 \
     zfcp.allow_lun_scan=0 \
     rd.dasd=0.0.3490
     ```

     将参数文件中的所有选项写为一行，并确保您没有换行符。

4. 将 initramfs、内核、参数文件和 RHCOS 镜像传送到 z/VM 中，例如使用 FTP。有关如何使用 FTP 传输文件并从虚拟 reader 引导的详情，请参考 在 Z/VM 中安装。

5. 将文件 punch 到 z/VM 虚拟机的虚拟 reader，即成为 bootstrap 节点。
   请参阅 IBM 文档中的 PUNCH。

提示

您可以使用 CP PUNCH 命令（如果是 Linux，使用 **vmur** 命令）在两个 z/VM 虚拟机间传输文件。

6. 在 bootstrap 机器中登录到 CMS。
7. 从 reader IPL bootstrap 机器：

```
$ ipl c
```

请参阅 IBM 文档中的 IPL。

8. 对集群中的其他机器重复此步骤。

1.2.8.1. 高级 RHCOS 安装参考

本节演示了网络配置和其他高级选项，允许您修改 Red Hat Enterprise Linux CoreOS（RHCOS）手动安装过程。下表描述了您可以与 RHCOS live installer 和 **coreos-installer** 命令一起使用的内核参数和命令行选项。

RHCOS 启动提示下的路由和绑定选项
如果从 ISO 镜像安装 RHCOS，您可以在引导该镜像时手动添加内核参数以配置节点的网络。如果没有使用网络参数，则安装默认为使用 DHCP。

重要

添加网络参数时，还必须添加 **rd.neednet=1** 内核参数。

下表描述了如何为实时 ISO 安装使用 **ip=** 、 **nameserver=** 和 **bond=** 内核参数。

注意

在添加内核参数时顺序非常重要： **ip=**，**nameserver=**，然后 **bond=**。

ISO 的路由和绑定选项

下表提供了配置 Red Hat Enterprise Linux CoreOS（RHCOS）节点网络的示例。这些是在系统引导过程中传递给 **dracut** 工具的网络选项。有关 **dracut** 支持的网络选项的详情，请参考 **dracut.cmdline** 手册页。

描述	例子
要配置一个 IP 地址，可以使用 DHCP(**ip=dhcp**)或者设置单独的静态 IP 地址(**ip=<host_ip>**)。然后在每个节点上指定 DNS 服务器 IP 地址(**nameserver=<dns_ip>**)。这个示例设置： • 节点的 IP 地址为 **10.10.10.2** • 网关地址为 **10.10.10.254** • 子网掩码为 **255.255.255.0** • 主机名为 **core0.example.com** • DNS 服务器地址为 **4.4.4.41**	ip=10.10.10.2::10.10.10.254:255.255.255.0:core0.example.com:enp1s0:none nameserver=4.4.4.41
通过指定多个 **ip=** 条目来指定多个网络接口。	ip=10.10.10.2::10.10.10.254:255.255.255.0:core0.example.com:enp1s0:none ip=10.10.10.3::10.10.10.254:255.255.255.0:core0.example.com:enp2s0:none
可选：您可以通过设置一个 **rd.route=** 值来配置到额外网络的路由。 如果额外网络网关与主要网络网关不同，则默认网关必须是主要网络网关。	配置默认网关： ip=::10.10.10.254:::: 为额外网络配置路由： rd.route=20.20.20.0/24:20.20.20.254:enp2s0
在单一接口中禁用 DHCP，比如当有两个或者多个网络接口时，且只有一个接口被使用。	ip=10.10.10.2::10.10.10.254:255.255.255.0:core0.example.com:enp1s0:none ip=::::core0.example.com:enp2s0:none
您可以将系统中 DHCP 和静态 IP 配置与多个网络接口结合在一起。	ip=enp1s0:dhcp ip=10.10.10.2::10.10.10.254:255.255.255.0:core0.example.com:enp2s0:none

描述	例子
可选： 您可以使用 **vlan=** 参数在单独的接口上配置 VLAN。	在网络接口中配置 VLAN 并使用静态 IP 地址： ip=10.10.10.2::10.10.10.254:255.255.255.0:core0.example.com:enp2s0.100:none vlan=enp2s0.100:enp2s0 在网络接口中配置 VLAN 并使用 DHCP： ip=enp2s0.100:dhcp vlan=enp2s0.100:enp2s0
您可以为每个服务器添加一个 **nameserver=** 条目来提供多个 DNS 服务器。	nameserver=1.1.1.1 nameserver=8.8.8.8
可选：使用 **bond=** 选项支持将多个网络接口绑定到一个接口。在这两个示例中： - 配置绑定接口的语法为：**bond=name[:network_interfaces][:options]** - *name* 是绑定设备名称（**bond0**），*network_interfaces* 代表用逗号分开的物理（以太网）接口(**em1,em2**)的列表，*options* 是用逗号分开的绑定选项列表。输入 **modinfo bonding** 查看可用选项。 - 当使用 **bond=** 创建绑定接口时，您必须指定如何分配 IP 地址以及绑定接口的其他信息。	要将绑定的接口配置为使用 DHCP，请将绑定的 IP 地址设置为 **dhcp**。例如： bond=bond0:em1,em2:mode=active-backup ip=bond0:dhcp 要将绑定接口配置为使用静态 IP 地址，请输入您需要的特定 IP 地址以及相关信息。例如： bond=bond0:em1,em2:mode=active-backup ip=10.10.10.2::10.10.10.254:255.255.255.0:core0.example.com:bond0:none
可选： 您可以使用 **vlan=** 参数在绑定接口上配置 VLAN。	使用 VLAN 配置绑定接口并使用 DHCP： ip=bond0.100:dhcp bond=bond0:em1,em2:mode=active-backup vlan=bond0.100:bond0 使用 VLAN 配置绑定接口，并使用静态 IP 地址： ip=10.10.10.2::10.10.10.254:255.255.255.0:core0.example.com:bond0.100:none bond=bond0:em1,em2:mode=active-backup vlan=bond0.100:bond0

描述	例子

可选：使用 **team=** 参数将网络团队用作绑定的替代选择。在本例中：
• 配置组接口的语法为：**team=name[:network_interfaces]** *name* 是团队设备名称（**team0**），*network_interfaces* 代表用逗号分开的物理（以太网）接口（**em1、em2**）。
 	
注意
当 RHCOS 切换到即将发布的 RHEL 版本时，团队计划被弃用。如需更多信息，请参阅 Red Hat 知识库文章。
	配置网络团队：
	`team=team0:em1,em2`
	`ip=team0:dhcp`

1.2.9. 创建集群

要创建 OpenShift Container Platform 集群，请等待您通过安装程序生成的 Ignition 配置文件所置备的机器上完成 bootstrap 过程。

先决条件

- 为集群创建所需的基础架构。
- 已获得安装程序并为集群生成了 Ignition 配置文件。
- 已使用 Ignition 配置文件为集群创建 RHCOS 机器。

流程

1. 监控 bootstrap 过程：

```
$ ./openshift-install --dir <installation_directory> wait-for bootstrap-complete \ 1
    --log-level=info 2
```

1 对于 **<installation_directory>**，请指定安装文件保存到的目录的路径。

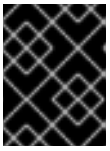
2 要查看不同的安装详情，请指定 **warn**、**debug** 或 **error**，而不要指定 **info**。

输出示例

```
INFO Waiting up to 30m0s for the Kubernetes API at https://api.test.example.com:6443...
INFO API v1.19.0 up
INFO Waiting up to 30m0s for bootstrapping to complete...
INFO It is now safe to remove the bootstrap resources
```

Kubernetes API 服务器提示已在 control plane 机器上完成 bootstrap 时，命令运行成功。

2. bootstrap 过程完成后，请从负载均衡器中删除 bootstrap 机器。



重要

此时您必须从负载均衡器中删除 bootstrap 机器。您还可以删除或重新格式化机器本身。

1.2.10. 使用 CLI 登录到集群

您可以通过导出集群 **kubeconfig** 文件，以默认系统用户身份登录集群。**kubeconfig** 文件包含关于集群的信息，供 CLI 用于将客户端连接到正确集群和 API 服务器。该文件特只适用于一个特定的集群，在 OpenShift Container Platform 安装过程中创建。

先决条件

- 已部署了 OpenShift Container Platform 集群。
- 已安装 **oc** CLI。

流程

1. 导出 **kubeadmin** 凭证：

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

1 对于 **<installation_directory>**，请指定安装文件保存到的目录的路径。

2. 使用导出的配置，验证能否成功运行 **oc** 命令：

```
$ oc whoami
```

输出示例

```
system:admin
```

1.2.11. 批准机器的证书签名请求

将机器添加到集群时，会为您添加的每台机器生成两个待处理证书签名请求（CSR）。您必须确认这些 CSR 已获得批准，或根据需要自行批准。客户端请求必须首先被批准，然后是服务器请求。

先决条件

- 您已将机器添加到集群中。

流程

1. 确认集群可以识别这些机器：

```
$ oc get nodes
```

输出示例

```
NAME     STATUS    ROLES   AGE  VERSION
master-0 Ready     master  63m  v1.19.0
```

```
master-1  Ready    master  63m  v1.19.0
master-2  Ready    master  64m  v1.19.0
```

输出将列出您创建的所有机器。

注意

在一些 CSR 被批准前，以上输出可能不包括计算节点（也称为 worker 节点）。

2. 检查待处理的 CSR，并确保可以看到添加到集群中的每台机器都有 **Pending** 或 **Approved** 状态的客户端请求：

```
$ oc get csr
```

输出示例

```
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
...
```

在本例中，两台机器加入了集群。您可能会在列表中看到更多已批准的 CSR。

3. 如果 CSR 没有获得批准，请在所添加机器的所有待处理 CSR 都处于 **Pending** 状态后，为您的集群机器批准这些 CSR：

注意

由于 CSR 会自动轮转，因此请在将机器添加到集群后一小时内批准您的 CSR。如果没有在一小时内批准，证书将会轮转，每个节点将会存在多个证书。您必须批准所有这些证书。批准客户端 CSR 后， Kubelet 为服务证书创建辅助 CSR，这需要手动批准。然后，如果 Kubelet 请求具有相同参数的新证书，则 **machine-approver** 会自动批准后续服务证书续订请求。

注意

对于在未启用机器 API 的平台中运行的集群，如裸机和其他用户置备的基础架构，必须采用一种方法自动批准 kubelet 提供证书请求（CSR）。如果没有批准请求，则 **oc exec**、**oc rsh** 和 **oc logs** 命令将无法成功，因为 API 服务器连接到 kubelet 时需要服务证书。与 Kubelet 端点联系的任何操作都需要此证书批准。这个方法必须监视新的 CSR，确认 CSR 由 **system:node** 或 **system:admin** 组中的 **node-bootstrapper** 服务帐户提交，并确认节点的身份。

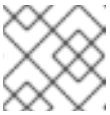
- 若要单独批准，请对每个有效的 CSR 运行以下命令：

```
$ oc adm certificate approve <csr_name> 1
```

1 **<csr_name>** 是当前 CSR 列表中 CSR 的名称。

- 要批准所有待处理的 CSR，请运行以下命令：

```
$ 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
```



注意

在有些 CSR 被批准前，一些 Operator 可能无法使用。

4. 现在，您的客户端请求已被批准，您必须查看添加到集群中的每台机器的服务器请求：

```
$ oc get csr
```

输出示例

```
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. 如果剩余的 CSR 没有被批准，且处于 **Pending** 状态，请批准集群机器的 CSR：

- 若要单独批准，请对每个有效的 CSR 运行以下命令：

```
$ oc adm certificate approve <csr_name> 1
```

1 **<csr_name>** 是当前 CSR 列表中 CSR 的名称。

- 要批准所有待处理的 CSR，请运行以下命令：

```
$ oc get csr -o go-template='{{range .items}}{{if not .status}}{{.metadata.name}}{{"\n"}}
{{end}}{{end}}' | xargs oc adm certificate approve
```

6. 批准所有客户端和服务器 CSR 后，器将处于 **Ready** 状态。运行以下命令验证：

```
$ oc get nodes
```

输出示例

```
NAME      STATUS    ROLES   AGE  VERSION
master-0  Ready     master  73m  v1.20.0
master-1  Ready     master  73m  v1.20.0
master-2  Ready     master  74m  v1.20.0
worker-0  Ready     worker  11m  v1.20.0
worker-1  Ready     worker  11m  v1.20.0
```

注意

批准服务器 CSR 后可能需要几分钟时间让机器转换为 **Ready** 状态。

其他信息

- 如需有关 CSR 的更多信息，请参阅证书签名请求。

1.2.12. 初始 Operator 配置

在 control plane 初始化后，您必须立即配置一些 Operator 以便它们都可用。

先决条件

- 您的 control plane 已初始化。

流程

1. 观察集群组件上线：

```
$ watch -n5 oc get clusteroperators
```

输出示例

```
NAME                                       VERSION AVAILABLE   PROGRESSING   DEGRADED
SINCE
authentication                             4.6.0   True        False         False      3h56m
cloud-credential                           4.6.0   True        False         False      29h
cluster-autoscaler                         4.6.0   True        False         False      29h
config-operator                            4.6.0   True        False         False      6h39m
console                                    4.6.0   True        False         False      3h59m
csi-snapshot-controller                    4.6.0   True        False         False      4h12m
dns                                        4.6.0   True        False         False      4h15m
etcd                                       4.6.0   True        False         False      29h
image-registry                             4.6.0   True        False         False      3h59m
ingress                                    4.6.0   True        False         False      4h30m
insights                                   4.6.0   True        False         False      29h
kube-apiserver                             4.6.0   True        False         False      29h
kube-controller-manager                    4.6.0   True        False         False      29h
kube-scheduler                             4.6.0   True        False         False      29h
kube-storage-version-migrator              4.6.0   True        False         False      4h2m
machine-api                                4.6.0   True        False         False      29h
machine-approver                           4.6.0   True        False         False      6h34m
machine-config                             4.6.0   True        False         False      3h56m
marketplace                                4.6.0   True        False         False      4h2m
monitoring                                 4.6.0   True        False         False      6h31m
network                                    4.6.0   True        False         False      29h
node-tuning                                4.6.0   True        False         False      4h30m
openshift-apiserver                        4.6.0   True        False         False      3h56m
openshift-controller-manager               4.6.0   True        False         False      4h36m
openshift-samples                          4.6.0   True        False         False      4h30m
operator-lifecycle-manager                 4.6.0   True        False         False      29h
operator-lifecycle-manager-catalog         4.6.0   True        False         False      29h
operator-lifecycle-manager-packageserver   4.6.0   True        False         False      3h59m
service-ca                                 4.6.0   True        False         False      29h
storage                                    4.6.0   True        False         False      4h30m
```

2. 配置不可用的 Operator。

1.2.12.1. 禁用默认的 OperatorHub 源

在 OpenShift Container Platform 安装过程中，默认为 OperatorHub 配置由红帽和社区项目提供的源内容的 operator 目录。在受限网络环境中，必须以集群管理员身份禁用默认目录。

流程

- 通过在 **OperatorHub** 对象中添加 **disableAllDefaultSources: true** 来禁用默认目录的源：

```
$ oc patch OperatorHub cluster --type json \
    -p '[{"op": "add", "path": "/spec/disableAllDefaultSources", "value": true}]'
```

提示

或者，您可以使用 Web 控制台管理目录源。在 **Administration → Cluster Settings → Global Configuration → OperatorHub** 页面中，点 **Sources** 选项卡，其中可创建、删除、禁用和启用单独的源。

1.2.12.2. 镜像 registry 存储配置

对于不提供默认存储的平台，Image Registry Operator 最初将不可用。安装后，您必须配置 registry 使用的存储，这样 Registry Operator 才可用。

示配置生产集群所需的持久性卷的说明。如果适用，显示有关将空目录配置为存储位置的说明，该位置只可用于非生产集群。

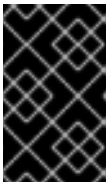
另外还提供了在升级过程中使用 **Recreate** rollout 策略来允许镜像 registry 使用块存储类型的说明。

1.2.12.2.1. 为 IBM Z 配置 registry 存储

作为集群管理员，在安装后需要配置 registry 来使用存储。

先决条件

- 具有 Cluster Administrator 权限
- IBM Z 上的集群。
- 为集群置备的持久性存储。



重要

如果您只有一个副本，OpenShift Container Platform 支持对镜像 registry 存储的 **ReadWriteOnce** 访问。要部署支持高可用性的、带有两个或多个副本的镜像 registry，需要 **ReadWriteMany** 访问设置。

- 必须具有 100Gi 容量。

流程

1. 为了配置 registry 使用存储，需要修改 **configs.imageregistry/cluster** 资源中的 **spec.storage.pvc** 。

注意

使用共享存储时，请查看您的安全设置以防止被外部访问。

2. 验证您没有 registry pod:

```
$ oc get pod -n openshift-image-registry
```

注意

如果存储类型为 **emptyDIR**，则副本数不能超过 **1**。

3. 检查 registry 配置：

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

输出示例

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

将 **claim** 字段留空以允许自动创建一个 **image-registry-storage** PVC。

4. 检查 **clusteroperator** 的状态：

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

5. 确保您的 registry 设置为 manage，以启用镜像的构建和推送。

- 运行：

```
$ oc edit configs.imageregistry/cluster
```

然后将行改

```
managementState: Removed
```

为

```
managementState: Managed
```

1.2.12.2.2. 在非生产集群中配置镜像 registry 存储

您必须为 Image Registry Operator 配置存储。对于非生产集群，您可以将镜像 registry 设置为空目录。如果您这样做，重启 registry 后会丢失所有镜像。

流程

- 将镜像 registry 存储设置为空目录：

```
$ oc patch configs.imageregistry.operator.openshift.io cluster --type merge --patch '{"spec":{"storage":{"emptyDir":{}}}}'
```

警告

仅可为非生产集群配置这个选项。

如果在 Image Registry Operator 初始化其组件前运行此命令，**oc patch** 命令会失败并显示以下错误：

```
Error from server (NotFound): configs.imageregistry.operator.openshift.io "cluster" not found
```

等待几分钟，然后再次运行该命令。

1.2.13. 在用户置备的基础架构上完成安装

完成 Operator 配置后，可以在您提供的基础架构上完成集群安装。

先决条件

- 您的 control plane 已初始化。
- 已完成初始 Operator 配置。

流程

1. 使用以下命令确认所有集群组件都已在线：

```
$ watch -n5 oc get clusteroperators
```

输出示例

```
NAME                                       VERSION AVAILABLE   PROGRESSING   DEGRADED
SINCE
authentication                             4.6.0   True        False         False      3h56m
cloud-credential                           4.6.0   True        False         False      29h
cluster-autoscaler                         4.6.0   True        False         False      29h
config-operator                            4.6.0   True        False         False      6h39m
console                                    4.6.0   True        False         False      3h59m
csi-snapshot-controller                    4.6.0   True        False         False      4h12m
dns                                        4.6.0   True        False         False      4h15m
etcd                                       4.6.0   True        False         False      29h
image-registry                             4.6.0   True        False         False      3h59m
ingress                                    4.6.0   True        False         False      4h30m
insights                                   4.6.0   True        False         False      29h
kube-apiserver                             4.6.0   True        False         False      29h
kube-controller-manager                    4.6.0   True        False         False      29h
kube-scheduler                             4.6.0   True        False         False      29h
```

```
kube-storage-version-migrator              4.6.0   True        False         False      4h2m
machine-api                                4.6.0   True        False         False      29h
machine-approver                           4.6.0   True        False         False      6h34m
machine-config                             4.6.0   True        False         False      3h56m
marketplace                                4.6.0   True        False         False      4h2m
monitoring                                 4.6.0   True        False         False      6h31m
network                                    4.6.0   True        False         False      29h
node-tuning                                4.6.0   True        False         False      4h30m
openshift-apiserver                        4.6.0   True        False         False      3h56m
openshift-controller-manager               4.6.0   True        False         False      4h36m
openshift-samples                          4.6.0   True        False         False      4h30m
operator-lifecycle-manager                 4.6.0   True        False         False      29h
operator-lifecycle-manager-catalog         4.6.0   True        False         False      29h
operator-lifecycle-manager-packageserver   4.6.0   True        False         False      3h59m
service-ca                                 4.6.0   True        False         False      29h
storage                                    4.6.0   True        False         False      4h30m
```

或者，通过以下命令，如果所有集群都可用您会接到通知。它还检索并显示凭证：

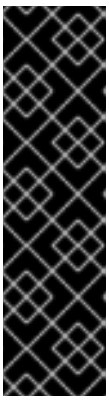
```
$ ./openshift-install --dir <installation_directory> wait-for install-complete 1
```

1 对于 **<installation_directory>**，请指定安装文件保存到的目录的路径。

输出示例

```
INFO Waiting up to 30m0s for the cluster to initialize...
```

Cluster Version Operator 完成从 Kubernetes API 服务器部署 OpenShift Container Platform 集群时，命令运行成功。



重要

- 安装程序生成的 Ignition 配置文件包含在 24 小时后过期的证书，然后在过期时进行续订。如果在更新证书前关闭集群，且集群在 24 小时后重启，集群会自动恢复过期的证书。一个例外情况是，您需要手动批准待处理的 **node-bootstrapper** 证书签名请求（CSR）来恢复 kubelet 证书。如需更多信息，请参阅*从过期的 control plane 证书中恢复*的文档。
- 建议您在生成 12 小时后使用 Ignition 配置文件，因为集群安装后 24 小时证书从 16 小时轮转至 22 小时。通过在 12 小时内使用 Ignition 配置文件，您可以避免在安装过程中运行证书更新时避免安装失败。

2. 确认 Kubernetes API 服务器正在与 pod 通信。

   a. 要查看所有 pod 的列表，请使用以下命令：

   ```
   $ oc get pods --all-namespaces
   ```

   输出示例

   ```
   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. 使用以下命令，查看上一命令的输出中所列 pod 的日志：

```
$ oc logs <pod_name> -n <namespace> 1
```

1 指定 pod 名称和命名空间，如上一命令的输出中所示。

如果 pod 日志显示，Kubernetes API 服务器可以与集群机器通信。

3. 在 Cluster registration 页面注册您的集群。

1.2.14. OpenShift Container Platform 的 Telemetry 访问

在 OpenShift Container Platform 4.6 中，默认运行的 Telemetry 服务提供有关集群健康状况和成功更新的指标，需要访问互联网。如果您的集群连接到互联网，Telemetry 会自动运行，而且集群会注册到 OpenShift Cluster Manager。

确认 OpenShift Cluster Manager 清单正确后，可以由 Telemetry 自动维护，也可以使用 OpenShift Cluster Manager 手动维护，使用订阅监控来跟踪帐户或多集群级别的 OpenShift Container Platform 订阅。

其他资源

- 有关 Telemetry 服务的更多信息，请参阅关于远程健康监控。

1.2.15. 收集调试信息

您可以收集有助于在 IBM Z 中安装 OpenShift Container Platform 时对特定问题进行故障排除和调试的调试信息。

先决条件

- 安装了 **oc** CLI 工具

流程

1. 登录到集群：

```
$ oc login
```

2. 在您要收集硬件信息的节点中，启动一个调试容器：

```
$ oc debug node/<nodename>
```

3. 进入 **/host** 文件系统并启动 **toolbox**:

```
$ chroot /host
$ toolbox
```

4. 收集 **dbginfo** 数据：

```
$ dbginfo.sh
```

5. 然后可以使用 **scp**来获取数据。

其他资源

- 请参阅在没有 SSH 的情况下在 OpenShift Container Platform 版本 4 中生成 SOSREPORT 。

1.2.16. 后续步骤

- 自定义集群。
- 如果您用来安装集群的镜像 registry 具有一个可信任的 CA，通过配置额外的信任存储将其添加到集群中。