



OpenShift Container Platform 4.6

在 OpenStack 上安装

安装 OpenShift Container Platform OpenStack 集群

OpenShift Container Platform 4.6 在 OpenStack 上安装

安装 OpenShift Container Platform OpenStack 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

Copyright © 2022 | You need to change the HOLDER entity in the en-US/Installing_on_OpenStack.ent file |.

The text of and illustrations in this document are licensed by Red Hat under a Creative Commons Attribution–Share Alike 3.0 Unported license ("CC-BY-SA"). An explanation of CC-BY-SA is available at

<http://creativecommons.org/licenses/by-sa/3.0/>

. In accordance with CC-BY-SA, if you distribute this document or an adaptation of it, you must provide the URL for the original version.

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

Red Hat, Red Hat Enterprise Linux, the Shadowman logo, the Red Hat logo, JBoss, OpenShift, Fedora, the Infinity logo, and RHCE are trademarks of Red Hat, Inc., registered in the United States and other countries.

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

Java[®] is a registered trademark of Oracle and/or its affiliates.

XFS[®] is a trademark of Silicon Graphics International Corp. or its subsidiaries in the United States and/or other countries.

MySQL[®] is a registered trademark of MySQL AB in the United States, the European Union and other countries.

Node.js[®] is an official trademark of Joyent. Red Hat is not formally related to or endorsed by the official Joyent Node.js open source or commercial project.

The OpenStack[®] Word Mark and OpenStack logo are either registered trademarks/service marks or trademarks/service marks of the OpenStack Foundation, in the United States and other countries and are used with the OpenStack Foundation's permission. We are not affiliated with, endorsed or sponsored by the OpenStack Foundation, or the OpenStack community.

All other trademarks are the property of their respective owners.

摘要

本文档提供在 OpenStack Container Platform 上安装和卸载 OpenShift Container Platform 集群的说明。

目录

第 1 章 在 OPENSTACK 上安装	6
1.1. 使用自定义配置在 OPENSTACK 上安装集群	6
1.1.1. 先决条件	6
1.1.2. 在 RHOSP 上安装 OpenShift Container Platform 的资源指南	6
1.1.2.1. control plane 机器	7
1.1.2.2. 计算机器	7
1.1.2.3. bootstrap 机器	7
1.1.3. OpenShift Container Platform 的互联网访问	8
1.1.4. 在 RHOSP 上启用 Swift	8
1.1.5. 验证外部网络访问	9
1.1.6. 为安装程序定义参数	10
1.1.7. 获取安装程序	11
1.1.8. 创建安装配置文件	12
1.1.8.1. 在安装过程中配置集群范围代理	13
1.1.9. 安装配置参数	14
1.1.9.1. 所需的配置参数	15
1.1.9.2. 网络配置参数	16
1.1.9.3. 可选配置参数	17
1.1.9.4. 其他 Red Hat OpenStack Platform (RHOSP) 配置参数	20
1.1.9.5. 可选 RHOSP 配置参数	21
1.1.9.6. RHOSP 部署中的自定义子网	24
1.1.9.7. RHOSP 的自定义 install-config.yaml 文件示例	24
1.1.10. 设置计算机器关联性	25
1.1.11. 生成 SSH 私钥并将其添加到代理中	27
1.1.12. 启用对环境的访问	28
1.1.12.1. 启用通过浮动 IP 地址进行访问	28
1.1.12.2. 完成没有浮动 IP 地址的安装	30
1.1.13. 部署集群	30
1.1.14. 验证集群状态	31
1.1.15. 使用 CLI 登录到集群	32
1.1.16. OpenShift Container Platform 的 Telemetry 访问	33
1.1.17. 后续步骤	33
1.2. 在带有 KURYR 的 OPENSTACK 上安装集群	33
1.2.1. 先决条件	33
1.2.2. 关于 Kuryr SDN	33
1.2.3. 在带有 Kuryr 的 OpenStack 上安装 OpenShift Container Platform 的资源指南	34
1.2.3.1. 增加配额	36
1.2.3.2. 配置 Neutron	36
1.2.3.3. 配置 Octavia	36
1.2.3.3.1. Octavia OVN 驱动程序	39
1.2.3.4. 已知使用 Kuryr 安装的限制	40
RHOSP 常规限制	40
RHOSP 版本限制	40
RHOSP 环境限制	40
RHOSP 升级限制	41
1.2.3.5. control plane 机器	41
1.2.3.6. 计算机器	41
1.2.3.7. bootstrap 机器	42
1.2.4. OpenShift Container Platform 的互联网访问	42
1.2.5. 在 RHOSP 上启用 Swift	42
1.2.6. 验证外部网络访问	43

1.2.7. 为安装程序定义参数	44
1.2.8. 获取安装程序	45
1.2.9. 创建安装配置文件	46
1.2.9.1. 在安装过程中配置集群范围代理	47
1.2.10. 安装配置参数	49
1.2.10.1. 所需的配置参数	49
1.2.10.2. 网络配置参数	50
1.2.10.3. 可选配置参数	51
1.2.10.4. 其他 Red Hat OpenStack Platform (RHOSP) 配置参数	54
1.2.10.5. 可选 RHOSP 配置参数	55
1.2.10.6. RHOSP 部署中的自定义子网	57
1.2.10.7. 使用 Kuryr 的 RHOSP 的自定义 install-config.yaml 文件示例	58
1.2.10.8. Kuryr 端口池	59
1.2.10.9. 在安装过程中调整 Kuryr 端口池	59
1.2.11. 设置计算机关联性	61
1.2.12. 生成 SSH 私钥并将其添加到代理中	63
1.2.13. 启用对环境的访问	64
1.2.13.1. 启用通过浮动 IP 地址进行访问	64
1.2.13.2. 完成没有浮动 IP 地址的安装	66
1.2.14. 部署集群	66
1.2.15. 验证集群状态	67
1.2.16. 使用 CLI 登录到集群	68
1.2.17. OpenShift Container Platform 的 Telemetry 访问	69
1.2.18. 后续步骤	69
1.3. 在您自己的基础架构的 OPENSTACK 上安装集群	69
1.3.1. 先决条件	69
1.3.2. OpenShift Container Platform 的互联网访问	69
1.3.3. 在 RHOSP 上安装 OpenShift Container Platform 的资源指南	70
1.3.3.1. control plane 机器	71
1.3.3.2. 计算机器	71
1.3.3.3. bootstrap 机器	71
1.3.4. 下载 playbook 的依赖项	71
1.3.5. 下载安装 playbook	72
1.3.6. 获取安装程序	73
1.3.7. 生成 SSH 私钥并将其添加到代理中	74
1.3.8. 创建 Red Hat Enterprise Linux CoreOS (RHCOS) 镜像	75
1.3.9. 验证外部网络访问	76
1.3.10. 启用对环境的访问	77
1.3.10.1. 启用通过浮动 IP 地址进行访问	77
1.3.10.2. 完成没有浮动 IP 地址的安装	78
1.3.11. 为安装程序定义参数	79
1.3.12. 创建安装配置文件	80
1.3.13. 安装配置参数	82
1.3.13.1. 所需的配置参数	82
1.3.13.2. 网络配置参数	83
1.3.13.3. 可选配置参数	84
1.3.13.4. 其他 Red Hat OpenStack Platform (RHOSP) 配置参数	87
1.3.13.5. 可选 RHOSP 配置参数	88
1.3.13.6. RHOSP 部署中的自定义子网	90
1.3.13.7. RHOSP 的自定义 install-config.yaml 文件示例	91
1.3.13.8. 为机器设置自定义子网	92
1.3.13.9. 清空计算机器池	92
1.3.14. 创建 Kubernetes 清单和 Ignition 配置文件	93

1.3.15. 准备 bootstrap Ignition 文件	94
1.3.16. 在 RHOSP 上创建 control plane Ignition 配置文件	97
1.3.17. 在 RHOSP 上创建网络资源	97
1.3.18. 在 RHOSP 上创建 bootstrap 机器	99
1.3.19. 在 RHOSP 中创建 control plane 机器	99
1.3.20. 使用 CLI 登录到集群	100
1.3.21. 从 RHOSP 删除 bootstrap 资源	101
1.3.22. 在 RHOSP 上创建计算机器	101
1.3.23. 批准机器的证书签名请求	102
1.3.24. 验证安装是否成功	104
1.3.25. OpenShift Container Platform 的 Telemetry 访问	105
1.3.26. 后续步骤	105
1.4. 在您自己的基础架构上带有 KURYR 的 OPENSTACK 上安装集群	105
1.4.1. 先决条件	105
1.4.2. 关于 Kuryr SDN	106
1.4.3. 在带有 Kuryr 的 OpenStack 上安装 OpenShift Container Platform 的资源指南	106
1.4.3.1. 增加配额	108
1.4.3.2. 配置 Neutron	108
1.4.3.3. 配置 Octavia	108
1.4.3.3.1. Octavia OVN 驱动程序	111
1.4.3.4. 已知使用 Kuryr 安装的限制	112
RHOSP 常规限制	112
RHOSP 版本限制	112
RHOSP 环境限制	113
RHOSP 升级限制	113
1.4.3.5. control plane 机器	113
1.4.3.6. 计算机器	114
1.4.3.7. bootstrap 机器	114
1.4.4. OpenShift Container Platform 的互联网访问	114
1.4.5. 下载 playbook 的依赖项	115
1.4.6. 下载安装 playbook	115
1.4.7. 获取安装程序	116
1.4.8. 生成 SSH 私钥并将其添加到代理中	117
1.4.9. 创建 Red Hat Enterprise Linux CoreOS (RHCOS) 镜像	118
1.4.10. 验证外部网络访问	119
1.4.11. 启用对环境的访问	120
1.4.11.1. 启用通过浮动 IP 地址进行访问	120
1.4.11.2. 完成没有浮动 IP 地址的安装	121
1.4.12. 为安装程序定义参数	122
1.4.13. 创建安装配置文件	123
1.4.14. 安装配置参数	125
1.4.14.1. 所需的配置参数	125
1.4.14.2. 网络配置参数	126
1.4.14.3. 可选配置参数	127
1.4.14.4. 其他 Red Hat OpenStack Platform (RHOSP) 配置参数	130
1.4.14.5. 可选 RHOSP 配置参数	131
1.4.14.6. RHOSP 部署中的自定义子网	133
1.4.14.7. 使用 Kuryr 的 RHOSP 的自定义 install-config.yaml 文件示例	134
1.4.14.8. Kuryr 端口池	135
1.4.14.9. 在安装过程中调整 Kuryr 端口池	135
1.4.14.10. 为机器设置自定义子网	137
1.4.14.11. 清空计算机器池	138
1.4.14.12. 修改网络类型	138

1.4.15. 创建 Kubernetes 清单和 Ignition 配置文件	139
1.4.16. 准备 bootstrap Ignition 文件	140
1.4.17. 在 RHOSP 上创建 control plane Ignition 配置文件	143
1.4.18. 在 RHOSP 上创建网络资源	143
1.4.19. 在 RHOSP 上创建 bootstrap 机器	145
1.4.20. 在 RHOSP 中创建 control plane 机器	145
1.4.21. 使用 CLI 登录到集群	146
1.4.22. 从 RHOSP 删除 bootstrap 资源	146
1.4.23. 在 RHOSP 上创建计算机	147
1.4.24. 批准机器的证书签名请求	148
1.4.25. 验证安装是否成功	150
1.4.26. OpenShift Container Platform 的 Telemetry 访问	150
1.4.27. 后续步骤	151
1.5. 在受限网络中的 OPENSTACK 上安装集群	151
1.5.1. 关于在受限网络中安装	151
1.5.1.1. 其他限制	152
1.5.2. 在 RHOSP 上安装 OpenShift Container Platform 的资源指南	152
1.5.2.1. control plane 机器	153
1.5.2.2. 计算机	153
1.5.2.3. bootstrap 机器	153
1.5.3. OpenShift Container Platform 的互联网访问	153
1.5.4. 在 RHOSP 上启用 Swift	154
1.5.5. 为安装程序定义参数	154
1.5.6. 为受限网络安装创建 RHCOS 镜像	156
1.5.7. 创建安装配置文件	157
1.5.7.1. 在安装过程中配置集群范围代理	159
1.5.7.2. 安装配置参数	160
1.5.7.2.1. 所需的配置参数	160
1.5.7.2.2. 网络配置参数	161
1.5.7.2.3. 可选配置参数	163
1.5.7.2.4. 其他 Red Hat OpenStack Platform (RHOSP) 配置参数	166
1.5.7.2.5. 可选 RHOSP 配置参数	167
1.5.7.3. 受限 OpenStack 安装的自定义 install-config.yaml 文件示例	170
1.5.8. 设置计算机关联性	171
1.5.9. 生成 SSH 私钥并将其添加到代理中	173
1.5.10. 启用对环境的访问	174
1.5.10.1. 启用通过浮动 IP 地址进行访问	174
1.5.10.2. 完成没有浮动 IP 地址的安装	175
1.5.11. 部署集群	176
1.5.12. 验证集群状态	177
1.5.13. 使用 CLI 登录到集群	178
1.5.14. 禁用默认的 OperatorHub 源	178
1.5.15. OpenShift Container Platform 的 Telemetry 访问	179
1.5.16. 后续步骤	179
1.6. 在 OPENSTACK 上卸载集群	179
1.6.1. 删除使用安装程序置备的基础架构的集群	179
1.7. 从您自己的基础架构中卸载 RHOSP 上的集群	180
1.7.1. 下载 playbook 的依赖项	180
1.7.2. 从使用您自己的基础架构的 RHOSP 中删除集群	181

第 1 章 在 OPENSTACK 上安装

1.1. 使用自定义配置在 OPENSTACK 上安装集群

在 OpenShift Container Platform 版本 4.6 中，您可以在 Red Hat OpenStack Platform (RHOSP) 上安装自定义集群。要自定义安装，请在安装集群前修改 `install-config.yaml` 中的参数。

1.1.1. 先决条件

- 查看有关 [OpenShift Container Platform 安装和更新](#) 流程的详细信息。
 - 在 *Available platforms* 部分验证 OpenShift Container Platform 4.6 是否与您的 RHOSP 版本兼容。您还可以查看 [OpenShift Container Platform 在 RHOSP 中的支持](#) 来比较不同版本的平台支持。
- 验证您的网络配置不依赖于供应商网络。不支持提供商网络。
- 在 RHOSP 中安装了存储服务，如块存储 (Cinder) 或对象存储 (Swift)。对象存储是 OpenShift Container Platform registry 集群部署的推荐存储技术。如需更多信息，请参阅 [优化存储](#)。
- 在 RHOSP 中启用了元数据服务

1.1.2. 在 RHOSP 上安装 OpenShift Container Platform 的资源指南

您的 Red Hat OpenStack Platform (RHOSP) 配额需要满足以下条件才支持 OpenShift Container Platform 安装：

表 1.1. RHOSP 上默认 OpenShift Container Platform 集群的建议资源

资源	值
浮动 IP 地址	3
端口	15
路由器	1
子网	1
RAM	112 GB
vCPUs	28
卷存储	275 GB
实例	7
安全组	3

资源	值
安全组规则	60

集群或许能使用少于推荐数量的资源来运作，但其性能无法保证。



重要

如果 RHOSP 对象存储 (Swift) 可用，并由具有 **swiftoperator** 角色的用户帐户执行，它会作为 OpenShift Container Platform 镜像 registry 的默认后端。在这种情况下，卷存储需要有 175GB。根据镜像 registry 的大小，Swift 空间要求会有所不同。



注意

默认情况下，您的安全组和安全组规则配额可能较低。如果遇到问题，请以 admin 的身份运行 `openstack quota set --secgroups 3 --secgroup-rules 60 <project>` 来提高配额。

OpenShift Container Platform 部署由 control plane 机器、计算机器和 bootstrap 机器组成。

1.1.2.1. control plane 机器

默认情况下，OpenShift Container Platform 安装过程会创建三台 control plane 机器。

每台机器都需要：

- 来自 RHOSP 配额的实例
- 来自 RHOSP 配额的端口
- 至少有 16 GB 内存、4 个 vCPU 和 100 GB 存储空间类别

1.1.2.2. 计算机器

默认情况下，OpenShift Container Platform 安装过程会创建三台计算机器。

每台机器都需要：

- 来自 RHOSP 配额的实例
- 来自 RHOSP 配额的端口
- 至少有 8 GB 内存、2 个 vCPU 和 100 GB 存储空间类别

提示

计算机器托管您在 OpenShift Container Platform 上运行的应用程序；运行数量应尽可能多。

1.1.2.3. bootstrap 机器

在安装时，会临时置备 bootstrap 机器来支持 control plane。生产控制平面就绪后，bootstrap 机器会被取消置备。

bootstrap 机器需要：

- 来自 RHOSP 配额的实例
- 来自 RHOSP 配额的端口
- 至少有 16 GB 内存、4 个 vCPU 和 100 GB 存储空间类别

1.1.3. 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。

1.1.4. 在 RHOSP 上启用 Swift

Swift 由具有 **swiftoperator** 角色的用户帐户操控。在运行安装程序前，将该角色添加到帐户。



重要

如果 [Red Hat OpenStack Platform \(RHOSP\) 对象存储服务](#) (通常称为 Swift) 可用，OpenShift Container Platform 会使用它作为镜像 registry 存储。如果无法使用，安装程序将依赖于 RHOSP 块存储服务，通常称为 Cinder。

如果 Swift 存在且您想要使用 Swift，则必须启用对其的访问。如果不存在，或者您不想使用它，请跳过这个部分。

先决条件

- 在目标环境中具有 RHOSP 管理员帐户
- 已安装 Swift 服务。
- 在 [Ceph RGW](#) 上启用了 **account in url** 选项。

流程

在 RHOSP 上启用 Swift：

1. 在 RHOSP CLI 中以管理员身份，将 **swiftoperator** 角色添加到要访问 Swift 的帐户：

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

您的 RHOSP 部署现可以使用 Swift 用于镜像 registry。

1.1.5. 验证外部网络访问

OpenShift Container Platform 安装进程需要外部网络访问权限。您必须为其提供外部网络值，否则部署会失败。在运行安装进程前，请验证 Red Hat OpenStack Platform (RHOSP) 中是否存在具有外部路由器类型的网络。

先决条件

- 将 OpenStack 联网服务配置为使用 DHCP 代理转发实例 DNS 查询

流程

1. 使用 RHOSP CLI 验证“外部”网络的名称和 ID：

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

输出示例

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

网络列表中会显示具有外部路由器类型的网络。如果最少有一个没有，请参阅 [创建默认浮动 IP 网络](#) 和 [创建默认供应商网络](#)。

重要

如果外部网络 CIDR 范围与某一个默认网络范围重叠，您必须在运行安装进程前更改 `install-config.yaml` 文件中匹配的网络范围。

默认的网络范围：

网络	范围
<code>machineNetwork</code>	10.0.0.0/16
<code>serviceNetwork</code>	172.30.0.0/16
<code>clusterNetwork</code>	10.128.0.0/14



警告

如果安装程序找到多个同名的镜像，它会随机设置其中之一。为避免这种行为，请在 RHOSP 中为资源创建唯一名称。



注意

如果启用了 Neutron 中继服务插件，则默认创建中继端口。如需更多信息，请参阅 [Neutron 中继端口](#)。

1.1.6. 为安装程序定义参数

OpenShift Container Platform 安装程序依赖于一个名为 **clouds.yaml** 的文件。该文件描述了 Red Hat OpenStack Platform (RHOSP) 配置参数，包括项目名称、登录信息和授权服务 URL。

流程

1. 创建 **clouds.yaml** 文件：

- 如果您的 RHOSP 发行版包含 Horizon web UI，请在该 UI 中生成 **clouds.yaml** 文件。



重要

请记住在 **auth** 字段中添加密码。您也可以把 secret 保存在 **clouds.yaml** 以外的一个独立的文件中。

- 如果您的 RHOSP 发行版不包含 Horizon Web UI，或者您不想使用 Horizon，请自行创建该文件。如需有关 **clouds.yaml** 的详细信息，请参阅 RHOSP 文档中的 [配置文件](#)。

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

- 如果您的 RHOSP 安装使用自签名证书颁发机构 (CA) 证书进行端点身份验证：
 - 将 CA 文件复制到您的机器中。

- b. 将机器添加到证书颁发机构信任捆绑包中：

```
$ sudo cp ca.crt.pem /etc/pki/ca-trust/source/anchors/
```

- c. 更新信任捆绑包：

```
$ sudo update-ca-trust extract
```

- d. 将 **cacerts** 键添加到 **clouds.yaml** 文件。该值必须是到 CA 证书的绝对路径，则其可以被非根用户访问：

```
clouds:
  shiftstack:
  ...
  cacert: "/etc/pki/ca-trust/source/anchors/ca.crt.pem"
```

提示

使用自定义 CA 证书运行安装程序后，您可以通过编辑 **cloud-provider-config** keymap 中的 **ca-cert.pem** 键的值来更新证书。在命令行中运行：

```
$ oc edit configmap -n openshift-config cloud-provider-config
```

3. 将 **clouds.yaml** 文件放在以下位置之一：

- OS_CLIENT_CONFIG_FILE** 环境变量的值
- 当前目录
- 特定于 Unix 的用户配置目录，如 `~/.config/openshift/clouds.yaml`
- 特定于 Unix 的站点配置目录，如 `/etc/openshift/clouds.yaml`
安装程序会按照以上顺序搜索 **clouds.yaml**。

1.1.7. 获取安装程序

在安装 OpenShift Container Platform 之前，将安装文件下载到本地计算机上。

先决条件

- 运行 Linux 或 macOS 的计算机，本地磁盘空间为 500 MB

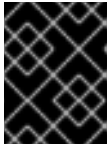
流程

- 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。
- 选择您的基础架构供应商。
- 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。

1.1.8. 创建安装配置文件

您可以自定义在 Red Hat OpenStack Platform (RHOSP) 上安装的 OpenShift Container Platform 集群。

先决条件

- 获取 OpenShift Container Platform 安装程序以及集群的 pull secret。

流程

1. 创建 `install-config.yaml` 文件。
 - a. 更改到包含安装程序的目录，再运行以下命令：

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

- 1** 对于 `<installation_directory>`，请指定用于保存安装程序所创建的文件目录名称。



重要

指定一个空目录。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。

- b. 在提示符处，提供您的云的配置详情：
 - i. 可选：选择用来访问集群机器的 SSH 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。

- ii. 选择 **openstack** 作为目标平台。
 - iii. 指定用于安装集群的 Red Hat OpenStack Platform (RHOSP) 外部网络名称。
 - iv. 指定用于从外部访问 OpenShift API 的浮动 IP 地址。
 - v. 指定至少有 16 GB RAM 用于 control plane 和计算节点的 RHOSP 类别。
 - vi. 选择集群要部署到的基域。所有 DNS 记录都将是这个基域的子域，并包含集群名称。
 - vii. 为集群输入一个名称。名称不能多于 14 个字符。
 - viii. 粘贴 [Red Hat OpenShift Cluster Manager 中的 pull secret](#)。
2. 修改 **install-config.yaml** 文件。您可以在[安装配置参数](#)部分中找到有关可用参数的更多信息。
 3. 备份 **install-config.yaml** 文件，以便用于安装多个集群。



重要

install-config.yaml 文件会在安装过程中消耗掉。如果要重复使用此文件，必须现在备份。

其他资源

[有关可用参数的更多信息](#)，请参阅[安装配置参数部分](#)。

1.1.8.1. 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

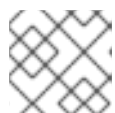
1. 编辑 `install-config.yaml` 文件并添加代理设置。例如：

```

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。

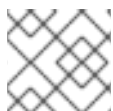


注意

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`。

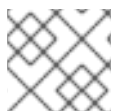


注意

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

1.1.9. 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。

1.1.9.1. 所需的配置参数

下表描述了所需的安装配置参数：

表 1.2. 所需的参数

参数	描述	值
<code>apiVersion</code>	<code>install-config.yaml</code> 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
<code>baseDomain</code>	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 <code>baseDomain</code> 和 <code>metadata.name</code> 参数值的组合，其格式为 <code><metadata.name>.<baseDomain></code> 。	完全限定域名或子域名，如 example.com 。
<code>metadata</code>	Kubernetes 资源 ObjectMeta ，其中只消耗 <code>name</code> 参数。	对象
<code>metadata.name</code>	集群的名称。集群的 DNS 记录是 <code>{{.metadata.name}}</code> 。 <code>{{.baseDomain}}</code> 的子域。	小写字母、连字符(-)和句点(.)的字符串，如 dev 。该字符串长度必须为 14 个字符或更少。
<code>platform</code>	执行安装的具体平台配置： aws 、 baremetal 、 azure 、 openstack 、 ovirt 、 vsphere 。有关 <code>platform.<platform></code> 参数的额外信息，请参考下表来了解您的具体平台。	对象

参数	描述	值
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>

1.1.9.2. 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 1.3. 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： <pre>networking: clusterNetwork: - cidr: 10.128.0.0/14 hostPrefix: 23</pre>
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。

参数	描述	值
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： <pre>networking: serviceNetwork: - 172.30.0.0/16</pre>
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。	一个对象数组。例如： <pre>networking: machineNetwork: - cidr: 10.0.0.0/16</pre>
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。

1.1.9.3. 可选配置参数

下表描述了可选安装配置参数：

表 1.4. 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。

参数	描述	值
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	<p>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。</p> <div style="display: flex; align-items: flex-start;">  <div> <p>重要</p> <p>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。</p> </div> </div>	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	<p>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。</p> <div style="display: flex; align-items: flex-start;">  <div> <p>重要</p> <p>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。</p> </div> </div>	Enabled 或 Disabled

参数	描述	值
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、o virt、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	<p>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。</p> <div style="display: flex; align-items: flex-start;"> <div style="width: 20px; height: 20px; background: repeating-linear-gradient(45deg, transparent, transparent 2px, black 2px, black 4px); margin-right: 10px;"></div> <div> <p>注意</p> <p>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 <i>Red Hat Operator 参考指南</i> 内容中的 <i>Cloud Credential Operator</i> 条目。</p> </div> </div>	Mint、Passthrough、Manual 或空字符串("")。
fips	<p>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。</p> <div style="display: flex; align-items: flex-start;"> <div style="width: 20px; height: 20px; background: repeating-linear-gradient(45deg, transparent, transparent 2px, black 2px, black 4px); margin-right: 10px; margin-bottom: 10px;"></div> <div> <p>重要</p> <p>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。</p> </div> </div> <div style="display: flex; align-items: flex-start;"> <div style="width: 20px; height: 20px; background: repeating-linear-gradient(45deg, transparent, transparent 2px, black 2px, black 4px); margin-right: 10px;"></div> <div> <p>注意</p> <p>如果使用 Azure File 存储，则无法启用 FIPS 模式。</p> </div> </div>	false 或 true

参数	描述	值
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。	<p>Internal 或 External。默认值为 External。</p> <p>在非云平台上不支持将此字段设置为 Internal。</p> <div style="display: flex; align-items: flex-start;">  <div> <p>重要</p> <p>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。</p> </div> </div>
sshKey	<p>用于验证集群机器访问的 SSH 密钥或密钥。</p> <div style="display: flex; align-items: flex-start;">  <div> <p>注意</p> <p>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。</p> </div> </div>	<p>一个或多个密钥。例如：</p> <pre>sshKey: <key1> <key2> <key3></pre>

1.1.9.4. 其他 Red Hat OpenStack Platform (RHOSP) 配置参数

下表描述了其他 RHOSP 配置参数：

表 1.5. 其他 RHOSP 参数

参数	描述	值
<code>compute.platform.openstack.rootVolume.size</code>	对于计算机器，以 GB 为单位表示的根卷大小。如果您不设置这个值，机器将使用临时存储。	整数，如 30 。
<code>compute.platform.openstack.rootVolume.type</code>	对于计算机器，根卷的类型。	字符串，如 performance 。
<code>controlPlane.platform.openstack.rootVolume.size</code>	对于 control plane 机器，以 GB 为单位表示的根卷大小。如果您不设置这个值，机器将使用临时存储。	整数，如 30 。
<code>controlPlane.platform.openstack.rootVolume.type</code>	对于 control plane 机器，根卷的类型。	字符串，如 performance 。
<code>platform.openstack.cloud</code>	要使用的 RHOSP 云的名称，来自于 <code>clouds.yaml</code> 文件中的云列表。	字符串，如 MyCloud 。
<code>platform.openstack.externalNetwork</code>	用于安装的 RHOSP 外部网络名称。	字符串，如 external 。
<code>platform.openstack.computeFlavor</code>	用于 control plane 和计算机器的 RHOSP 类别。	字符串，如 m1.xlarge 。

1.1.9.5. 可选 RHOSP 配置参数

下表描述了可选 RHOSP 配置参数：

表 1.6. 可选的 RHOSP 参数

参数	描述	值
<code>compute.platform.openstack.additionalNetworks</code>	与计算机器关联的其他网络。不能为额外网络创建允许的地址对。	一个或多个 UUID 列表作为字符串。例如： fa806b2f-ac49-4bce-b9db-124bc64209bf 。

参数	描述	值
compute.platform.openstack.additionalSecurityGroupIDs	与计算机器关联的其他安全组。	一个或多个 UUID 列表作为字符串。例如： 7ee219f3-d2e9-48a1-96c2-e7429f1b0da7 。
compute.platform.openstack.zones	<p>RHOSP Compute (Nova) 可用区 (AZ) 在其中安装机器。如果没有设置此参数，安装程序会依赖于配置了 RHOSP 管理员的 Nova 的默认设置。</p> <p>在使用 Kuryr 的集群上，RHOSP Octavia 不支持可用域。负载均衡器，如果您使用 Amphora 供应商驱动程序，则依赖 Amphora 虚拟机的 OpenShift Container Platform 服务不会根据此属性的值创建。</p>	字符串列表。例如： ["zone-1", "zone-2"] 。
controlPlane.platform.openstack.additionalNetworkIDs	与 control plane 机器关联的额外网络。不能为额外网络创建允许的地址对。	一个或多个 UUID 列表作为字符串。例如： fa806b2f-ac49-4bce-b9db-124bc64209bf 。
controlPlane.platform.openstack.additionalSecurityGroupIDs	与 control plane 机器关联的其他安全组。	一个或多个 UUID 列表作为字符串。例如： 7ee219f3-d2e9-48a1-96c2-e7429f1b0da7 。
controlPlane.platform.openstack.zones	<p>RHOSP Compute (Nova) 可用区 (AZ) 在其中安装机器。如果没有设置此参数，安装程序会依赖于配置了 RHOSP 管理员的 Nova 的默认设置。</p> <p>在使用 Kuryr 的集群上，RHOSP Octavia 不支持可用域。负载均衡器，如果您使用 Amphora 供应商驱动程序，则依赖 Amphora 虚拟机的 OpenShift Container Platform 服务不会根据此属性的值创建。</p>	字符串列表。例如： ["zone-1", "zone-2"] 。

参数	描述	值
platform.openstack.clusterOSImage	<p>安装程序从中下载 RHCOS 镜像的位置。</p> <p>您必须设置此参数以便在受限网络中执行安装。</p>	<p>HTTP 或 HTTPS URL，可选使用 SHA-256 checksum。</p> <p>例如： http://mirror.example.com/images/rhcos-43.81.201912131630.0-openstack.x86_64.qcow2.gz?sha256=ffebbd68e8a1f2a245ca19522c16c86f67f9ac8e4e0c1f0a812b068b16f7265d。该值也可以是现有 Glance 镜像的名称，如 my-rhcos。</p>
platform.openstack.defaultMachinePlatform	默认机器池平台配置。	<pre>{ "type": "ml.large", "rootVolume": { "size": 30, "type": "performance" } }</pre>
platform.openstack.ingressFloatingIP	<p>与 Ingress 端口关联的现有浮动 IP 地址。要使用此属性，还必须定义 platform.openstack.externalNetwork 属性。</p>	IP 地址，如 128.0.0.1 。
platform.openstack.lbFloatingIP	<p>与 API 负载均衡器关联的现有浮动 IP 地址。要使用此属性，还必须定义 platform.openstack.externalNetwork 属性。</p>	IP 地址，如 128.0.0.1 。
platform.openstack.externalDNS	集群实例用于进行 DNS 解析的外部 DNS 服务器的 IP 地址。	一个 IP 地址列表作为字符串。例如， ["8.8.8.8", "192.168.1.12"] 。

参数	描述	值
platform.openstack.machinesSubnet	<p>集群节点使用的 RHOSP 子网的 UUID。在这个子网上创建节点和虚拟 IP (VIP) 端口。</p> <p>networking.machineNetwork 中的第一个项需要和 machinesSubnet 的值匹配。</p> <p>如果部署到自定义子网中，则无法将外部 DNS 服务器指定到 OpenShift Container Platform 安装程序。反之，把 DNS 添加到 RHOSP 的子网。</p>	作为字符串的 UUID。例如： fa806b2f-ac49-4bceb9db-124bc64209bf 。

1.1.9.6. RHOSP 部署中的自定义子网

另外，您还可以在您选择的 Red Hat OpenStack Platform (RHOSP) 子网中部署集群。子网的 GUID 作为 **install-config.yaml** 文件中的 **platform.openstack.machinesSubnet** 的值传递。

此子网被用作集群的主子网，在其上创建节点和端口。

在使用自定义子网运行 OpenShift Container Platform 安装程序前，请验证：

- 目标网络和子网可用。
- 目标子网上启用了 DHCP。
- 您可提供在目标网络上有创建端口权限的安装程序凭证。
- 如果您的网络配置需要一个路由器，它会在 RHOSP 中创建。有些配置依赖于路由器来转换浮动 IP 地址。
- 您的网络配置不依赖于供应商网络。不支持提供商网络。

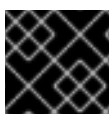


注意

默认情况下，API VIP 使用 x.x.x.5，Ingress VIP 从网络 CIDR 块获取 x.x.x.7。要覆盖这些默认值，为 DHCP 分配池以外的 **platform.openstack.apiVIP** 和 **platform.openstack.ingressVIP** 设置值。

1.1.9.7. RHOSP 的自定义 **install-config.yaml** 文件示例

此示例 **install-config.yaml** 展示了所有可能的 Red Hat OpenStack Platform (RHOSP) 自定义选项。



重要

此示例文件仅供参考。您必须使用安装程序来获取 **install-config.yaml** 文件。

apiVersion: v1
baseDomain: example.com

```

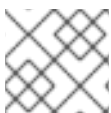
clusterID: os-test
controlPlane:
  name: master
  platform: {}
  replicas: 3
compute:
- name: worker
  platform:
    openstack:
      type: m1.large
      replicas: 3
metadata:
  name: example
networking:
  clusterNetwork:
  - cidr: 10.128.0.0/14
    hostPrefix: 23
  machineNetwork:
  - cidr: 10.0.0.0/16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

```

1.1.10. 设置计算机器关联性

另外，您还可以在安装过程中为计算机器设置关联性策略。默认情况下，安装程序不会为计算机器选择关联性策略。

您还可以在安装后创建使用特定 RHOSP 服务器组的机器集。



注意

control plane 机器使用 **soft-anti-affinity** 策略创建。

提示

您可以在 RHOSP 文档中了解更多有关 [RHOSP 实例调度和放置](#) 的信息。

先决条件

- 创建 **install-config.yaml** 文件并完成对其所做的任何修改。

流程

1. 使用 RHOSP 命令行界面，为您的计算机器创建服务器组。例如：

-

```
$ openstack \
  --os-compute-api-version=2.15 \
  server group create \
  --policy anti-affinity \
  my-openshift-worker-group
```

如需更多信息，请参阅[服务器组 create 命令文档](#)。

2. 进入包含安装程序的目录并创建清单：

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

其中：

installation_directory

指定包含集群的 **install-config.yaml** 文件的目录名称。

3. 打开 **manifests/99_openshift-cluster-api_worker-machineset-0.yaml**，这是 **MachineSet** 定义文件。
4. 将属性 **serverGroupID** 添加到 **spec.template.spec.providerSpec.value** 属性下的定义中。例如：

```
apiVersion: machine.openshift.io/v1beta1
kind: MachineSet
metadata:
  labels:
    machine.openshift.io/cluster-api-cluster: <infrastructure_ID>
    machine.openshift.io/cluster-api-machine-role: <node_role>
    machine.openshift.io/cluster-api-machine-type: <node_role>
  name: <infrastructure_ID>-<node_role>
  namespace: openshift-machine-api
spec:
  replicas: <number_of_replicas>
  selector:
    matchLabels:
      machine.openshift.io/cluster-api-cluster: <infrastructure_ID>
      machine.openshift.io/cluster-api-machineset: <infrastructure_ID>-<node_role>
  template:
    metadata:
      labels:
        machine.openshift.io/cluster-api-cluster: <infrastructure_ID>
        machine.openshift.io/cluster-api-machine-role: <node_role>
        machine.openshift.io/cluster-api-machine-type: <node_role>
        machine.openshift.io/cluster-api-machineset: <infrastructure_ID>-<node_role>
    spec:
      providerSpec:
        value:
          apiVersion: openstackproviderconfig.openshift.io/v1alpha1
          cloudName: openstack
          cloudsSecret:
            name: openstack-cloud-credentials
            namespace: openshift-machine-api
          flavor: <nova_flavor>
          image: <glance_image_name_or_location>
```

```

serverGroupID: aaaaaaaaa-bbbb-cccc-dddd-eeeeeeeeeeee 1
kind: OpenstackProviderSpec
networks:
- filter: {}
  subnets:
  - filter:
    name: <subnet_name>
    tags: openshiftClusterID=<infrastructure_ID>
securityGroups:
- filter: {}
  name: <infrastructure_ID>-<node_role>
serverMetadata:
  Name: <infrastructure_ID>-<node_role>
  openshiftClusterID: <infrastructure_ID>
tags:
- openshiftClusterID=<infrastructure_ID>
trunk: true
userDataSecret:
  name: <node_role>-user-data
availabilityZone: <optional_openstack_availability_zone>

```

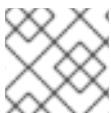
1 在此处添加服务器组的 UUID。

5. 可选：备份 **manifests/99_openshift-cluster-api_worker-machineset-0.yaml** 文件。创建集群时，安装程序会删除 **manifests/** 目录。

安装集群时，安装程序将使用您修改的 **MachineSet** 定义在 RHOSP 服务器组中创建计算机。

1.1.11. 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。

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。

1.1.12. 启用对环境的访问

在部署时，所有 OpenShift Container Platform 机器都是在 Red Hat OpenStack Platform (RHOSP) 租户网络中创建的。因此，大多数 RHOSP 部署中都无法直接访问它们。

您可以在安装过程中使用浮动 IP 地址（FIP）来配置 OpenShift Container Platform API 和应用程序访问。您也可以在没有配置 FIP 的情况下完成安装，但安装程序不会配置一种从外部访问 API 或应用程序的方法。

1.1.12.1. 启用通过浮动 IP 地址进行访问

创建浮动 IP（FIP）地址，用于从外部访问 OpenShift Container Platform API 和集群应用程序。

流程

1. 使用 Red Hat OpenStack Platform (RHOSP) CLI，创建 API FIP：


```
$ openstack floating ip create --description "API <cluster_name>.<base_domain>"
<external_network>
```

2. 使用 Red Hat OpenStack Platform (RHOSP) CLI, 创建应用程序或 Ingress, FIP :

```
$ openstack floating ip create --description "Ingress <cluster_name>.<base_domain>"
<external_network>
```

3. 向用于 API 和 Ingress FIP 的 DNS 服务器添加符合这些模式的记录 :

```
api.<cluster_name>.<base_domain>. IN A <API_FIP>
*.apps.<cluster_name>.<base_domain>. IN A <apps_FIP>
```

注意

如果您不控制 DNS 服务器, 您可以通过将集群域名 (如以下内容) 添加到 `/etc/hosts` 文件中来访问集群 :

- `<api_floating_ip> api.<cluster_name>.<base_domain>`
- `<application_floating_ip> grafana-openshift-monitoring.apps.<cluster_name>.<base_domain>`
- `<application_floating_ip> prometheus-k8s-openshift-monitoring.apps.<cluster_name>.<base_domain>`
- `<application_floating_ip> oauth-openshift.apps.<cluster_name>.<base_domain>`
- `<application_floating_ip> console-openshift-console.apps.<cluster_name>.<base_domain>`
- `application_floating_ip integrate-oauth-server-openshift-authentication.apps.<cluster_name>.<base_domain>`

`/etc/hosts` 文件中的集群域名授予对本地集群的 Web 控制台和监控界面的访问权限。您还可以使用 `kubectl` 或 `oc`。您可以使用指向 `<application_floating_ip>` 的额外条目来访问用户应用程序。此操作使 API 和应用程序可供您访问, 不适用于生产部署, 但允许对开发和测试进行安装。

4. 将 FIP 添加到 `install-config.yaml` 文件, 将其作为以下参数的值 :

- `platform.openstack.ingressFloatingIP`
- `platform.openstack.lbFloatingIP`

如果使用这些值, 还必须在 `install-config.yaml` 文件中输入一个外部网络作为 `platform.openstack.externalNetwork` 参数的值。

提示

您可以通过分配浮动 IP 地址并更新防火墙配置, 使 OpenShift Container Platform 资源在集群之外可用。

1.1.12.2. 完成没有浮动 IP 地址的安装

您可以在不提供浮动 IP 地址的情况下在 Red Hat OpenStack Platform (RHOSP) 上安装 OpenShift Container Platform。

在 `install-config.yaml` 文件中，不要定义以下参数：

- `platform.openstack.ingressFloatingIP`
- `platform.openstack.lbFloatingIP`

如果您无法提供外部网络，也可以将 `platform.openstack.externalNetwork` 留空。如果没有为 `platform.openstack.externalNetwork` 提供值，则不会为您创建路由器。如果没有额外的操作，安装程序将无法从 Glance 检索镜像。您必须自行配置外部连接。

如果在因为缺少浮动 IP 地址或名称解析而无法访问集群 API 的系统中运行安装程序时，安装会失败。要防止安装失败，可以使用代理网络或者从与您的机器位于同一网络的系统中运行安装程序。



注意

您可以通过为 API 和 Ingress 端口创建 DNS 记录来启用名称解析。例如：

```
api.<cluster_name>.<base_domain>. IN A <api_port_IP>
*.apps.<cluster_name>.<base_domain>. IN A <ingress_port_IP>
```

如果您不控制 DNS 服务器，可以改为将记录添加到 `/etc/hosts` 文件中。此操作使 API 可供您自己访问，不适用于生产部署。这可用于进行开发和测试的安装。

1.1.13. 部署集群

您可以在兼容云平台中安装 OpenShift Container Platform。



重要

安装程序的 `create cluster` 命令只能在初始安装过程中运行一次。

先决条件

- 获取 OpenShift Container Platform 安装程序以及集群的 pull secret。

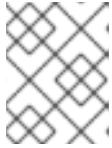
流程

1. 更改为包含安装程序的目录并初始化集群部署：

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

1 对于 `<installation_directory>`，请指定自定义 `./install-config.yaml` 文件的位置。

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



注意

如果您在主机上配置的云供应商帐户没有足够的权限来部署集群，安装过程将会停止，并且显示缺少的权限。

集群部署完成后，终端会显示访问集群的信息，包括指向其 Web 控制台的链接和 **kubeadmin** 用户的凭证。

输出示例

```

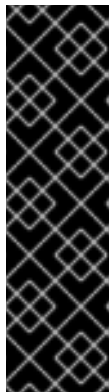
...
INFO Install complete!
INFO To access the cluster as the system:admin user when using 'oc', run 'export
KUBECONFIG=/home/myuser/install_dir/auth/kubeconfig'
INFO Access the OpenShift web-console here: https://console-openshift-
console.apps.mycluster.example.com
INFO Login to the console with user: "kubeadmin", and password: "4vYBz-Ee6gm-ymBZj-
Wt5AL"
INFO Time elapsed: 36m22s

```



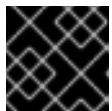
注意

当安装成功时，集群访问和凭证信息还会输出到 **<installation_directory>/openshift_install.log**。



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-bootstrap** 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。



重要

您不得删除安装程序或安装程序所创建的文件。需要这两者才能删除集群。

1.1.14. 验证集群状态

您可以在安装过程中或安装后验证 OpenShift Container Platform 集群的状态：

流程

1. 在集群环境中，导出管理员的 kubeconfig 文件：

```
$ 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。

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。

2. 查看部署后创建的 control plane 和计算机器：

```
$ oc get nodes
```

3. 查看集群的版本：

```
$ oc get clusterversion
```

4. 查看 Operator 的状态：

```
$ oc get clusteroperator
```

5. 查看集群中的所有正在运行的 pod:

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

1.1.15. 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
```

其他资源

- 如需有关访问和了解 OpenShift Container Platform Web 控制台的更多信息，请参阅[访问 Web 控制台](#)。

1.1.16. 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](#)。

1.1.17. 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](#)。
- 如果您需要启用对节点端口的外部访问，[请使用节点端口配置集群流量](#)。
- 如果您没有将 RHOSP 配置为使用浮动 IP 地址接受应用程序流量，[使用浮动 IP 地址配置 RHOSP 访问](#)。

1.2. 在带有 KURYR 的 OPENSTACK 上安装集群

在 OpenShift Container Platform 版本 4.6 中，您可以在使用 Kuryr SDN 的 Red Hat OpenStack Platform (RHOSP) 上安装自定义集群。要自定义安装，请在安装集群前修改 `install-config.yaml` 中的参数。

1.2.1. 先决条件

- 查看有关 [OpenShift Container Platform 安装和更新](#) 流程的详细信息。
 - 在 *Available platforms* 部分验证 OpenShift Container Platform 4.6 是否与您的 RHOSP 版本兼容。您还可以查看 [OpenShift Container Platform 在 RHOSP 中的支持](#) 来比较不同版本的平台支持。
- 验证您的网络配置不依赖于供应商网络。不支持提供商网络。
- 在 RHOSP 中安装了存储服务，如块存储 (Cinder) 或对象存储 (Swift)。对象存储是 OpenShift Container Platform registry 集群部署的推荐存储技术。如需更多信息，请参阅 [优化存储](#)。

1.2.2. 关于 Kuryr SDN

[Kuryr](#) 是一个容器网络接口 (CNI) 插件解决方案，它使用 [Neutron](#) 和 [Octavia](#) Red Hat OpenStack Platform (RHOSP) 服务来为 pod 和服务提供网络。

Kuryr 和 OpenShift Container Platform 的集成主要针对在 RHOSP VM 上运行的 OpenShift Container Platform 集群设计。Kuryr 通过将 OpenShift Container Platform pod 插入到 RHOSP SDN 来提高网络性能。另外，它还提供 pod 和 RHOSP 虚拟实例间的互联性。

Kuryr 组件作为 pod 在 OpenShift Container Platform 中安装，使用 `openshift-kuryr` 命名空间：

- **kuryr-controller** - 在一个 **master** 节点上安装的单个服务实例。这在 OpenShift Container Platform 中建模为一个 **Deployment** 对象。
- **kuryr-cni** - 在每个 OpenShift Container Platform 节点上安装并配置 Kuryr 作为 CNI 驱动的容器。这在 OpenShift Container Platform 中建模为一个 **DaemonSet** 对象。

Kuryr 控制器监控 OpenShift Container Platform API 服务器中的 pod、服务和命名空间创建、更新和删除事件。它将 OpenShift Container Platform API 调用映射到 Neutron 和 Octavia 中的对应对象。这意味着，实现了 Neutron 中继端口功能的每个网络解决方案都可以通过 Kuryr 支持 OpenShift Container Platform。这包括开源解决方案，比如 Open vSwitch (OVS) 和 Open Virtual Network (OVN)，以及 Neutron 兼容的商业 SDN。

建议在封装的 RHOSP 租户网络上部署 OpenShift Container Platform 时使用 Kuryr，以避免出现重复封装，例如通过 RHOSP 网络运行封装的 OpenShift Container Platform SDN。

如果您使用供应商网络或租户 VLAN，则不需要使用 Kuryr 来避免重复封装。虽然性能上的优势微不足道，但根据您的配置，使用 Kuryr 避免两个覆盖可能仍然有用。

在完足以下所有条件的部署中不建议使用 Kuryr：

- RHOSP 版本早于 16
- 部署使用 UDP 服务，或者在几个 hypervisor 上使用大量 TCP 服务。

或

- **ovn-octavia** Octavia 驱动被禁用。
- 部署在几个 hypervisor 中使用了大量的 TCP 服务。

1.2.3. 在带有 Kuryr 的 OpenStack 上安装 OpenShift Container Platform 的资源指南

当使用 Kuryr SDN 时，pod、服务、命名空间和网络策略会使用来自 RHOSP 配额的资源，这会增加最低要求。除了默认安装需要满足的要求，Kuryr 还有一些额外的要求。

使用以下配额来满足集群的默认最低要求：

表 1.7. 带有 Kuryr 的 RHOSP 上默认 OpenShift Container Platform 集群的建议资源

资源	值
浮动 IP 地址	3 - 加上预期的 LoadBalancer 类型服务的数量
端口	1500 - 每个 Pod 需要 1 个
路由器	1
子网	250 - 每个命名空间/项目需要 1 个
网络	250 - 每个命名空间/项目需要 1 个
RAM	112 GB

资源	值
vCPUs	28
卷存储	275 GB
实例	7
安全组	250 - 每个服务和每个 NetworkPolicy 需要 1 个
安全组规则	1000
负载均衡器	100 - 每个服务需要 1 个
负载均衡器侦听程序	500 - 每个服务公开端口需要 1 个
负载均衡器池	500 - 每个服务公开端口需要 1 个

集群或许能使用少于推荐数量的资源来运作，但其性能无法保证。



重要

如果 RHOSP 对象存储 (Swift) 可用，并由具有 **swiftoperator** 角色的用户帐户执行，它会作为 OpenShift Container Platform 镜像 registry 的默认后端。在这种情况下，卷存储需要有 175GB。根据镜像 registry 的大小，Swift 空间要求会有所不同。



重要

如果您使用带有 Amphora 驱动而不是 OVN Octavia 驱动的 Red Hat OpenStack Platform (RHOSP) 版本 16，则安全组会与 service 帐户而不是用户项目关联。

在设置资源时请考虑以下几点：

- 需要的端口数量会大于 pod 的数量。Kuryr 使用端口池来预创建端口以供 pod 使用，用于加快 pod 的启动时间。
- 每个网络策略都映射到 RHOSP 安全组中，并根据 **NetworkPolicy** 规格将一个或多个规则添加到安全组中。
- 每个服务都映射到一个 RHOSP 负载均衡器中。在估算配额所需安全组数时，请考虑此要求。如果您使用 RHOSP 版本 15 或更早版本，或者使用 **ovn-octavia** 驱动，则每个负载均衡器都有一个带有用户项目的安全组。
- 配额不考虑负载均衡器资源（如 VM 资源），但您必须在决定 RHOSP 部署的大小时考虑这些资源。默认安装将有超过 50 个负载均衡器，集群必须可以容纳它们。如果您使用启用 OVN Octavia 驱动程序的 RHOSP 版本 16，则只生成一个负载均衡器虚拟机；服务通过 OVN 流平衡负载。

OpenShift Container Platform 部署由 control plane 机器、计算机器和 bootstrap 机器组成。

要启用 Kuryr SDN，您的环境必须满足以下要求：

- 运行 RHOSP 13+。
- 具有 Octavia 的 Overcloud。
- 使用 Neutron Trunk 端口扩展。
- 如果使用 ML2/OVS Neutron 驱动而不是 **ovs-hybrid**，则请使用 **openvswitch** 防火墙驱动。

1.2.3.1. 增加配额

使用 Kuryr SDN 时，您必须提高配额以满足 pod、Services、namespaces 和网络策略所使用的 Red Hat OpenStack Platform (RHOSP) 资源要求。

流程

- 运行以下命令为项目增加配额：

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

1.2.3.2. 配置 Neutron

Kuryr CNI 利用 Neutron Trunks 扩展来将容器插入 Red Hat OpenStack Platform (RHOSP) SDN，因此您必须使用 **trunks** 扩展才可以使 Kuryr 正常工作。

另外，如果您使用默认的 ML2/OVS Neutron 驱动程序，防火墙必须设为 **openvswitch** 而不是 **ovs_hybrid**，以便在中继子端口上强制实施安全组，同时 Kuryr 可以正确处理网络策略。

1.2.3.3. 配置 Octavia

Kuryr SDN 使用 Red Hat OpenStack Platform (RHOSP) 的 Octavia LBaaS 来实现 OpenShift Container Platform 服务。因此，您必须在 RHOSP 上安装和配置 Octavia 组件以使用 Kuryr SDN。

要启用 Octavia，您必须在安装 RHOSP Overcloud 的过程中包括 Octavia 服务，如果 Overcloud 已存在则需要升级 Octavia 服务。以下启用 Octavia 的步骤适用于新的 Overcloud 安装或 Overcloud 更新。



注意

以下步骤只包括在部署 RHOSP 时需要处理 Octavia 部分的信息。请注意 [registry](#) 可能会不同。

这个示例使用本地的 registry。

流程

1. 如果您使用本地 registry，请创建一个模板来将镜像上传到 registry。例如：

```
(undercloud) $ openstack overcloud container image prepare \
-e /usr/share/openstack-tripleo-heat-templates/environments/services-docker/octavia.yaml \
--namespace=registry.access.redhat.com/rhosp13 \
--push-destination=<local-ip-from-undercloud.conf>:8787 \
--prefix=openstack-
```



```
--tag-from-label {version}-{release} \
--output-env-file=/home/stack/templates/overcloud_images.yaml \
--output-images-file /home/stack/local_registry_images.yaml
```

- 验证 `local_registry_images.yaml` 文件是否包含 Octavia 镜像。例如：

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



注意

Octavia 容器版本根据所安装的特定 RHOSP 版本的不同而有所不同。

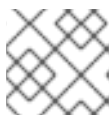
- 将 `registry.redhat.io` 中的容器镜像拉取到 Undercloud 节点：

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

这可能需要一些时间，具体要看您的网络速度和 Undercloud 使用的磁盘。

- 由于 Octavia 负载均衡器是用来访问 OpenShift Container Platform API，所以您必须增加它们的监听程序的默认超时时间。默认超时为 50 秒。通过将以下文件传递给 Overcloud deploy 命令，将超时时间增加到 20 分钟：

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



注意

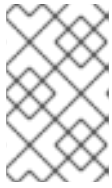
RHOSP 13.0.13+ 不需要这一步。

- 使用 Octavia 安装或更新 overcloud 环境：

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

**注意**

这个命令只包含与 Octavia 相关的文件，它根据您具体的 RHOSP 安装而有所不同。如需更多信息，请参阅 RHOSP 文档。有关自定义 Octavia 安装的详情请参考 [使用 Director 安装 Octavia](#)。

**注意**

当利用 Kuryr SDN 时，Overcloud 安装需要 Neutron **trunk** 扩展。这在 director 部署中默认可用。当 Neutron 后端是 ML2/OVS 时，使用 **openvswitch** 防火墙而不是默认的 **ovs-hybrid**。如果后端为 ML2/OVN，则不需要修改。

6. 在早于 13.0.13 的 RHOSP 版本中，在创建项目后将项目 ID 添加到 **octavia.conf** 配置文件中。

- 要跨服务实施网络策略，比如网络流量会通过 Octavia 负载均衡器时，您必须确保 Octavia 在用户项目中创建 Amphora VM 安全组。这可确保所需的 LoadBalancer 安全组属于该项目，并可将其更新为强制实施服务隔离。

**注意**

在 RHOSP 13.0.13 或更高版本中不需要此操作。

Octavia 实施新的 ACL API，限制对负载均衡器 VIP 的访问。

a. 获取项目 ID

```
$ openstack project show <project>
```

输出示例

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

b. 将项目 ID 添加到控制器的 **octavia.conf** 中。

i. Source **stackrc** 文件：

```
$ source stackrc # Undercloud credentials
```

ii. 列出 Overcloud 控制器。

```
$ openstack server list
```

输出示例

```

+-----+-----+-----+-----+
+-----+
|
| ID              | Name      | Status | Networks
| Image          | Flavor   |
|
+-----+-----+-----+-----+
+-----+
|
| 6bef8e73-2ba5-4860-a0b1-3937f8ca7e01 | controller-0 | ACTIVE |
ctlplane=192.168.24.8 | overcloud-full | controller |
|
| dda3173a-ab26-47f8-a2dc-8473b4a67ab9 | compute-0   | ACTIVE |
ctlplane=192.168.24.6 | overcloud-full | compute  |
|
+-----+-----+-----+-----+
+-----+

```

iii. SSH 到控制器。

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

iv. 编辑 **octavia.conf** 文件，将项目添加到 Amphora 安全组存在于用户账户的项目列表中。

```

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

```

c. 重启 Octavia worker 以便重新加载配置。

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



注意

根据您的 RHOSP 环境，Octavia 可能不支持 UDP 侦听程序。如果您在 RHOSP 版本 13.0.13 或更早版本使用 Kuryr SDN，则不支持 UDP 服务。RHOSP 版本 16 或更高版本支持 UDP。

1.2.3.3.1. Octavia OVN 驱动程序

Octavia 通过 Octavia API 支持多个供应商驱动程序。

要查看所有可用的 Octavia 提供程序驱动，请在命令行中输入：

```
$ openstack loadbalancer provider list
```

输出示例

```

+-----+-----+
| name | description |

```

```
+-----+-----+
| amphora | The Octavia Amphora driver.           |
| octavia | Deprecated alias of the Octavia Amphora driver. |
| ovn     | Octavia OVN driver.                       |
+-----+-----+
```

从 RHOSP 版本 16 开始，Octavia OVN 供应商驱动程序 (**ovn**) 在 RHOSP 部署的 OpenShift Container Platform 上被支持。

ovn 是 Octavia 和 OVN 提供的负载均衡集成驱动。它支持基本负载均衡功能，并基于 OpenFlow 规则。在使用 OVN Neutron ML2 的部署中，Director 会在 Octavia 中自动启用该驱动程序。

Amphora 供应商驱动程序是默认驱动程序。如果启用了 **ovn**，Kuryr 将使用它。

如果 Kuryr 使用 **ovn** 而不是 Amphora，则可提供以下优点：

- 资源要求更低 Kuryr 不需要为每个服务都提供一个负载均衡器虚拟机。
- 网络延迟会降低。
- 通过对每个服务使用 OpenFlow 规则而不是 VM 来提高服务创建速度。
- 跨所有节点的分布式负载均衡操作，而不是集中到 Amphora 虚拟机中。

您可以在 RHOSP 云从版本 13 升级到版本 16 后，[将集群配置为使用 Octavia OVN 驱动程序](#)。

1.2.3.4. 已知使用 Kuryr 安装的限制

将 OpenShift Container Platform 与 Kuryr SDN 搭配使用有一些已知的限制。

RHOSP 常规限制

带有 Kuryr SDN 的 OpenShift Container Platform 不支持带有类型 **NodePort** 的 **Service** 对象。

如果机器子网没有连接到路由器，或者子网已连接，但路由器没有设置外部网关，Kuryr 无法为类型为 **LoadBalancer** 的 **Service** 对象创建浮动 IP。

- 在 **Service** 对象上配置 **sessionAffinity=ClientIP** 属性无效。Kuryr 不支持此设置。

RHOSP 版本限制

使用带有 Kuryr SDN 的 OpenShift Container Platform 有一些限制，具体取决于 RHOSP 版本。

- RHOSP 16 之前的版本使用默认 Octavia 负载均衡器驱动程序 (Amphora)。此驱动要求在每个 OpenShift Container Platform 服务中部署一个 Amphora 负载均衡器虚拟机。创建太多的服务会导致您耗尽资源。
如果以后版本的 RHOSP 部署中禁用了 OVN Octavia 驱动程序，则也会使用 Amphora 驱动。它们对资源的要求和早期版本 RHOSP 相同。
- Octavia RHOSP 13.0.13 之前的版本不支持 UDP 侦听程序。因此，OpenShift Container Platform UDP 服务不被支持。
- Octavia RHOSP 13.0.13 之前的版本无法侦听同一端口上的多个协议。不支持将同一端口暴露给不同协议的服务，比如 TCP 和 UDP。
- Kuryr SDN 不支持由服务自动取消闲置。

RHOSP 环境限制

使用取决于您的部署环境的 Kuryr SDN 会有一些限制。

由于 Octavia 缺少对 UDP 协议和多个监听器的支持，如果 rhosp 版本早于 13.0.13，Kuryr 会强制 pod 在 DNS 解析中使用 TCP，如果：

在 Go 版本 1.12 及更早的版本中，通过 CGO 支持被禁用的模式编译的应用程序只使用 UDP。在这种情况下，native Go 解析器无法识别 **resolv.conf** 中的 **use-vc** 选项，它控制 DNS 解析是否强制使用 TCP。因此，UDP 仍会被用来解析 DNS，这将导致失败。

要确保 TCP 强制使用是允许的，在编译应用程序使把环境变量 **CGO_ENABLED** 设定为 **1**（如 **CGO_ENABLED=1**），或者不使用这个变量。

在 Go 版本 1.13 及之后的版本中，如果使用 UDP 的 DNS 解析失败，则会自动使用 TCP。



注意

基于 musl 的容器，包括基于 Alpine 的容器，不支持 **use-vc** 选项。

RHOSP 升级限制

作为 RHOSP 升级过程的结果，可能会更改 Octavia API，并可能需要升级到用于负载均衡器的 Amphora 镜像。

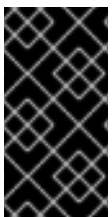
您可以单独处理 API 更改。

如果升级了 Amphora 镜像，RHOSP Operator 可使用两种方式处理现有的负载均衡器虚拟机：

- 通过触发[负载均衡器故障切换](#)来升级每个虚拟机。
- 将升级虚拟机的职责留给用户。

如果运算符使用第一个选项，在故障切换过程中可能会有短暂的停机时间。

如果 Operator 采用第二个选项，现有负载均衡器将不支持升级的 Octavia API 功能，比如 UDP 侦听程序。在这种情况下，用户必须重新创建自己的服务以使用这些功能。



重要

如果 OpenShift Container Platform 检测到支持 UDP 负载均衡的新 Octavia 版本，它会自动重新创建 DNS 服务。服务重新创建可确保服务默认支持 UDP 负载均衡。

这个重新创建会导致 DNS 服务大约停机一 分钟。

1.2.3.5. control plane 机器

默认情况下，OpenShift Container Platform 安装过程会创建三台 control plane 机器。

每台机器都需要：

- 来自 RHOSP 配额的实例
- 来自 RHOSP 配额的端口
- 至少有 16 GB 内存、4 个 vCPU 和 100 GB 存储空间类别

1.2.3.6. 计算机器

默认情况下，OpenShift Container Platform 安装过程会创建三台计算机器。

每台机器都需要：

- 来自 RHOSP 配额的实例
- 来自 RHOSP 配额的端口
- 至少有 8 GB 内存、2 个 vCPU 和 100 GB 存储空间类别

提示

计算机器托管您在 OpenShift Container Platform 上运行的应用程序；运行数量应尽可能多。

1.2.3.7. bootstrap 机器

在安装时，会临时置备 bootstrap 机器来支持 control plane。生产控制平面就绪后，bootstrap 机器会被取消置备。

bootstrap 机器需要：

- 来自 RHOSP 配额的实例
- 来自 RHOSP 配额的端口
- 至少有 16 GB 内存、4 个 vCPU 和 100 GB 存储空间类别

1.2.4. 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。

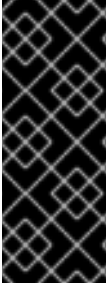


重要

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。

1.2.5. 在 RHOSP 上启用 Swift

Swift 由具有 **swiftoperator** 角色的用户帐户操控。在运行安装程序前，将该角色添加到帐户。



重要

如果 [Red Hat OpenStack Platform \(RHOSP\) 对象存储服务](#) (通常称为 Swift) 可用, OpenShift Container Platform 会使用它作为镜像 registry 存储。如果无法使用, 安装程序将依赖于 RHOSP 快存储服务, 通常称为 Cinder。

如果 Swift 存在且您想要使用 Swift, 则必须启用对其的访问。如果不存在, 或者您不想使用它, 请跳过这个部分。

先决条件

- 在目标环境中具有 RHOSP 管理员帐户
- 已安装 Swift 服务。
- 在 [Ceph RGW](#) 上启用了 **account in url** 选项。

流程

在 RHOSP 上启用 Swift :

1. 在 RHOSP CLI 中以管理员身份, 将 **swiftoperator** 角色添加到要访问 Swift 的帐户 :

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

您的 RHOSP 部署现可以使用 Swift 用于镜像 registry。

1.2.6. 验证外部网络访问

OpenShift Container Platform 安装进程需要外部网络访问权限。您必须为其提供外部网络值, 否则部署会失败。在运行安装进程前, 请验证 Red Hat OpenStack Platform (RHOSP) 中是否存在具有外部路由器类型的网络。

先决条件

- 将 [OpenStack 联网服务配置为使用 DHCP 代理转发实例 DNS 查询](#)

流程

1. 使用 RHOSP CLI 验证“外部”网络的名称和 ID :

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

输出示例

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

网络列表中会显示具有外部路由器类型的网络。如果最少有一个没有, 请参阅 [创建默认浮动 IP 网络](#)和 [创建默认供应商网络](#)。

重要

如果外部网络 CIDR 范围与某一个默认网络范围重叠，您必须在运行安装进程前更改 `install-config.yaml` 文件中匹配的网络范围。

默认的网络范围：

网络	范围
<code>machineNetwork</code>	10.0.0.0/16
<code>serviceNetwork</code>	172.30.0.0/16
<code>clusterNetwork</code>	10.128.0.0/14



警告

如果安装程序找到多个同名的镜像，它会随机设置其中之一。为避免这种行为，请在 RHOSP 中为资源创建唯一名称。



注意

如果启用了 Neutron 中继服务插件，则默认创建中继端口。如需更多信息，请参阅 [Neutron 中继端口](#)。

1.2.7. 为安装程序定义参数

OpenShift Container Platform 安装程序依赖于一个名为 `clouds.yaml` 的文件。该文件描述了 Red Hat OpenStack Platform (RHOSP) 配置参数，包括项目名称、登录信息和授权服务 URL。

流程

1. 创建 `clouds.yaml` 文件：

- 如果您的 RHOSP 发行版包含 Horizon web UI，请在该 UI 中生成 `clouds.yaml` 文件。



重要

请记住在 `auth` 字段中添加密码。您也可以把 secret 保存在 `clouds.yaml` 以外的一个独立的文件中。

- 如果您的 RHOSP 发行版不包含 Horizon Web UI，或者您不想使用 Horizon，请自行创建该文件。如需有关 `clouds.yaml` 的详细信息，请参阅 RHOSP 文档中的 [配置文件](#)。

```
clouds:
  shiftstack:
    auth:
```



```

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

```

2. 如果您的 RHOSP 安装使用自签名证书颁发机构 (CA) 证书进行端点身份验证：

- a. 将 CA 文件复制到您的机器中。
- b. 将机器添加到证书颁发机构信任捆绑包中：

```
$ sudo cp ca.crt.pem /etc/pki/ca-trust/source/anchors/
```

- c. 更新信任捆绑包：

```
$ sudo update-ca-trust extract
```

- d. 将 **cacerts** 键添加到 **clouds.yaml** 文件。该值必须是到 CA 证书的绝对路径，则其可以被非根用户访问：

```

clouds:
  shiftstack:
    ...
    cacert: "/etc/pki/ca-trust/source/anchors/ca.crt.pem"

```

提示

使用自定义 CA 证书运行安装程序后，您可以通过编辑 **cloud-provider-config** keymap 中的 **ca-cert.pem** 键的值来更新证书。在命令行中运行：

```
$ oc edit configmap -n openshift-config cloud-provider-config
```

3. 将 **clouds.yaml** 文件放在以下位置之一：

- a. **OS_CLIENT_CONFIG_FILE** 环境变量的值
- b. 当前目录
- c. 特定于 Unix 的用户配置目录，如 **~/.config/openstack/clouds.yaml**
- d. 特定于 Unix 的站点配置目录，如 **/etc/openstack/clouds.yaml**
安装程序会按照以上顺序搜索 **clouds.yaml**。

1.2.8. 获取安装程序

在安装 OpenShift Container Platform 之前，将安装文件下载到本地计算机上。

先决条件

- 运行 Linux 或 macOS 的计算机，本地磁盘空间为 500 MB

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。

1.2.9. 创建安装配置文件

您可以自定义在 Red Hat OpenStack Platform (RHOSP) 上安装的 OpenShift Container Platform 集群。

先决条件

- 获取 OpenShift Container Platform 安装程序以及集群的 pull secret。

流程

1. 创建 `install-config.yaml` 文件。
 - a. 更改到包含安装程序的目录，再运行以下命令：

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

- 1** 对于 `<installation_directory>`，请指定用于保存安装程序所创建的文件目录名称。



重要

指定一个空目录。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。

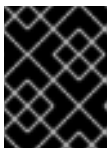
- b. 在提示符处，提供您的云的配置详情：
 - i. 可选：选择用来访问集群机器的 SSH 密钥。



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。

- ii. 选择 **openstack** 作为目标平台。
 - iii. 指定用于安装集群的 Red Hat OpenStack Platform (RHOSP) 外部网络名称。
 - iv. 指定用于从外部访问 OpenShift API 的浮动 IP 地址。
 - v. 指定至少有 16 GB RAM 用于 control plane 和计算节点的 RHOSP 类别。
 - vi. 选择集群要部署到的基域。所有 DNS 记录都将是这个基域的子域，并包含集群名称。
 - vii. 为集群输入一个名称。名称不能多于 14 个字符。
 - viii. 粘贴 [Red Hat OpenShift Cluster Manager 中的 pull secret](#)。
2. 修改 **install-config.yaml** 文件。您可以在 **安装配置参数** 部分中找到有关可用参数的更多信息。
 3. 备份 **install-config.yaml** 文件，以便用于安装多个集群。



重要

install-config.yaml 文件会在安装过程中消耗掉。如果要重复使用此文件，必须现在备份。

1.2.9.1. 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。

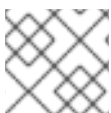


注意

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。

1.2.10. 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。

1.2.10.1. 所需的配置参数

下表描述了所需的安装配置参数：

表 1.8. 所需的参数

参数	描述	值
<code>apiVersion</code>	<code>install-config.yaml</code> 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
<code>baseDomain</code>	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 <code>baseDomain</code> 和 <code>metadata.name</code> 参数值的组合，其格式为 <code><metadata.name>.<baseDomain></code> 。	完全限定域名或子域名，如 example.com 。
<code>metadata</code>	Kubernetes 资源 ObjectMeta ，其中只消耗 <code>name</code> 参数。	对象
<code>metadata.name</code>	集群的名称。集群的 DNS 记录是 <code>{{.metadata.name}}</code> 。 <code>{{.baseDomain}}</code> 的子域。	小写字母、连字符(-)和句点(.)的字符串，如 dev 。该字符串长度必须为 14 个字符或更少。
<code>platform</code>	执行安装的具体平台配置： aws 、 baremetal 、 azure 、 openstack 、 ovirt 、 vsphere 。有关 <code>platform.<platform></code> 参数的额外信息，请参考下表来了解您的具体平台。	对象

参数	描述	值
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>

1.2.10.2. 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 1.9. 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： <pre>networking: clusterNetwork: - cidr: 10.128.0.0/14 hostPrefix: 23</pre>
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。

参数	描述	值
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： <pre>networking: serviceNetwork: - 172.30.0.0/16</pre>
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。	一个对象数组。例如： <pre>networking: machineNetwork: - cidr: 10.0.0.0/16</pre>
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 。  <p>注意</p> <p>将 networking.machineNetwork 设置为与首选 NIC 所在的 CIDR 匹配。</p>

1.2.10.3. 可选配置参数

下表描述了可选安装配置参数：

表 1.10. 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。

参数	描述	值
compute.architecture	决定池中机器的指令集架构。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	<p>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。</p>  <p>重要</p> <p>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。</p>	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 、 o virt 、 vsphere 或 {}
compute.replicas	要置备的计算器数量，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	<p>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。</p>  <p>重要</p> <p>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。</p>	Enabled 或 Disabled

参数	描述	值
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	<p>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。</p> <div style="display: flex; align-items: flex-start;"> <div style="width: 40px; height: 40px; background: repeating-linear-gradient(45deg, transparent, transparent 2px, #ccc 2px, #ccc 4px); margin-right: 10px;"></div> <div> <p>注意</p> <p>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 <i>Red Hat Operator 参考指南</i> 内容中的 <i>Cloud Credential Operator</i> 条目。</p> </div> </div>	Mint、Passthrough、Manual 或空字符串(“”)。
fips	<p>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。</p> <div style="display: flex; align-items: flex-start;"> <div style="width: 40px; height: 40px; background: repeating-linear-gradient(45deg, transparent, transparent 2px, #000 2px, #000 4px); margin-right: 10px; margin-bottom: 10px;"></div> <div> <p>重要</p> <p>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。</p> </div> </div> <div style="display: flex; align-items: flex-start;"> <div style="width: 40px; height: 40px; background: repeating-linear-gradient(45deg, transparent, transparent 2px, #ccc 2px, #ccc 4px); margin-right: 10px;"></div> <div> <p>注意</p> <p>如果使用 Azure File 存储，则无法启用 FIPS 模式。</p> </div> </div>	false 或 true

参数	描述	值
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。	<p>Internal 或 External。默认值为 External。</p> <p>在非云平台上不支持将此字段设置为 Internal。</p> <div style="display: flex; align-items: flex-start;"> <div style="width: 20px; height: 20px; background: repeating-linear-gradient(45deg, transparent, transparent 2px, black 2px, black 4px); margin-right: 10px;"></div> <div> <p>重要</p> <p>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。</p> </div> </div>
sshKey	用于验证集群机器访问的 SSH 密钥或密钥。 <div style="display: flex; align-items: flex-start; margin-top: 10px;"> <div style="width: 20px; height: 20px; background: repeating-linear-gradient(45deg, transparent, transparent 2px, black 2px, black 4px); margin-right: 10px;"></div> <div> <p>注意</p> <p>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。</p> </div> </div>	一个或多个密钥。例如： <pre style="margin-top: 10px;">sshKey: <key1> <key2> <key3></pre>

1.2.10.4. 其他 Red Hat OpenStack Platform (RHOSP) 配置参数

下表描述了其他 RHOSP 配置参数：

表 1.11. 其他 RHOSP 参数

参数	描述	值
<code>compute.platform.openstack.rootVolume.size</code>	对于计算机器，以 GB 为单位表示的根卷大小。如果您不设置这个值，机器将使用临时存储。	整数，如 30 。
<code>compute.platform.openstack.rootVolume.type</code>	对于计算机器，根卷的类型。	字符串，如 performance 。
<code>controlPlane.platform.openstack.rootVolume.size</code>	对于 control plane 机器，以 GB 为单位表示的根卷大小。如果您不设置这个值，机器将使用临时存储。	整数，如 30 。
<code>controlPlane.platform.openstack.rootVolume.type</code>	对于 control plane 机器，根卷的类型。	字符串，如 performance 。
<code>platform.openstack.cloud</code>	要使用的 RHOSP 云的名称，来自于 <code>clouds.yaml</code> 文件中的云列表。	字符串，如 MyCloud 。
<code>platform.openstack.externalNetwork</code>	用于安装的 RHOSP 外部网络名称。	字符串，如 external 。
<code>platform.openstack.computeFlavor</code>	用于 control plane 和计算机器的 RHOSP 类别。	字符串，如 m1.xlarge 。

1.2.10.5. 可选 RHOSP 配置参数

下表描述了可选 RHOSP 配置参数：

表 1.12. 可选的 RHOSP 参数

参数	描述	值
<code>compute.platform.openstack.additionalNetworkIDs</code>	与计算机器关联的其他网络。不能为额外网络创建允许的地址对。	一个或多个 UUID 列表作为字符串。例如： fa806b2f-ac49-4bce-b9db-124bc64209bf 。
<code>compute.platform.openstack.additionalSecurityGroupIDs</code>	与计算机器关联的其他安全组。	一个或多个 UUID 列表作为字符串。例如： 7ee219f3-d2e9-48a1-96c2-e7429f1b0da7 。

参数	描述	值
compute.platform.openstack.zones	<p>RHOSP Compute (Nova) 可用区 (AZ) 在其中安装机器。如果没有设置此参数, 安装程序会依赖于配置了 RHOSP 管理员的 Nova 的默认设置。</p> <p>在使用 Kuryr 的集群上, RHOSP Octavia 不支持可用域。负载均衡器, 如果您使用 Amphora 供应商驱动程序, 则依赖 Amphora 虚拟机的 OpenShift Container Platform 服务不会根据此属性的值创建。</p>	字符串列表。例如: ["zone-1", "zone-2"]。
controlPlane.platform.openstack.additionalNetworkIDs	与 control plane 机器关联的额外网络。不能为额外网络创建允许的地址对。	一个或多个 UUID 列表作为字符串。例如: fa806b2f-ac49-4bce-b9db-124bc64209bf 。
controlPlane.platform.openstack.additionalSecurityGroupIDs	与 control plane 机器关联的其他安全组。	一个或多个 UUID 列表作为字符串。例如: 7ee219f3-d2e9-48a1-96c2-e7429f1b0da7 。
controlPlane.platform.openstack.zones	<p>RHOSP Compute (Nova) 可用区 (AZ) 在其中安装机器。如果没有设置此参数, 安装程序会依赖于配置了 RHOSP 管理员的 Nova 的默认设置。</p> <p>在使用 Kuryr 的集群上, RHOSP Octavia 不支持可用域。负载均衡器, 如果您使用 Amphora 供应商驱动程序, 则依赖 Amphora 虚拟机的 OpenShift Container Platform 服务不会根据此属性的值创建。</p>	字符串列表。例如: ["zone-1", "zone-2"]。
platform.openstack.clusterOSImage	<p>安装程序从中下载 RHCOS 镜像的位置。</p> <p>您必须设置此参数以便在受限网络中执行安装。</p>	<p>HTTP 或 HTTPS URL, 可选使用 SHA-256 checksum。</p> <p>例如: http://mirror.example.com/images/rhcos-43.81.201912131630.0-openstack.x86_64.qcow2.gz?sha256=ffebbd68e8a1f2a245ca19522c16c86f67f9ac8e4e0c1f0a812b068b16f7265d。该值也可以是现有 Glance 镜像的名称, 如 my-rhcos。</p>

参数	描述	值
platform.openstack.defaultMachinePlatform	默认机器池平台配置。	<pre>{ "type": "ml.large", "rootVolume": { "size": 30, "type": "performance" } }</pre>
platform.openstack.ingressFloatingIP	与 Ingress 端口关联的现有浮动 IP 地址。要使用此属性，还必须定义 platform.openstack.externalNetwork 属性。	IP 地址，如 128.0.0.1 。
platform.openstack.lbFloatingIP	与 API 负载均衡器关联的现有浮动 IP 地址。要使用此属性，还必须定义 platform.openstack.externalNetwork 属性。	IP 地址，如 128.0.0.1 。
platform.openstack.externalDNS	集群实例用于进行 DNS 解析的外部 DNS 服务器的 IP 地址。	一个 IP 地址列表作为字符串。例如， ["8.8.8.8", "192.168.1.12"] 。
platform.openstack.machinesSubnet	<p>集群节点使用的 RHOSP 子网的 UUID。在这个子网上创建节点和虚拟 IP (VIP) 端口。</p> <p>networking.machineNetwork 中的第一个项需要和 machinesSubnet 的值匹配。</p> <p>如果部署到自定义子网中，则无法将外部 DNS 服务器指定到 OpenShift Container Platform 安装程序。反之，把 DNS 添加到 RHOSP 的子网。</p>	作为字符串的 UUID。例如： fa806b2f-ac49-4bceb9db-124bc64209bf 。

1.2.10.6. RHOSP 部署中的自定义子网

另外，您还可以在您选择的 Red Hat OpenStack Platform (RHOSP) 子网中部署集群。子网的 GUID 作为 **install-config.yaml** 文件中的 **platform.openstack.machinesSubnet** 的值传递。

此子网被用作集群的主子网，在其上创建节点和端口。

在使用自定义子网运行 OpenShift Container Platform 安装程序前，请验证：

- 目标网络和子网可用。

- 目标子网上启用了 DHCP。
- 您可提供在目标网络上有创建端口权限的安装程序凭证。
- 如果您的网络配置需要一个路由器，它会在 RHOSP 中创建。有些配置依赖于路由器来转换浮动 IP 地址。
- 您的网络配置不依赖于供应商网络。不支持提供商网络。

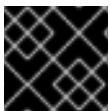


注意

默认情况下，API VIP 使用 x.x.x.5，Ingress VIP 从网络 CIDR 块获取 x.x.x.7。要覆盖这些默认值，为 DHCP 分配池以外的 **platform.openstack.apiVIP** 和 **platform.openstack.ingressVIP** 设置值。

1.2.10.7. 使用 Kuryr 的 RHOSP 的自定义 `install-config.yaml` 文件示例

要使用 Kuryr SDN 而不是默认的 OpenShift SDN 部署，您必须修改 `install-config.yaml` 文件，使其包含 **Kuryr** 作为所需的 **networking.networkType**，然后执行默认的 OpenShift Container Platform SDN 安装步骤。此示例 `install-config.yaml` 展示了所有可能的 Red Hat OpenStack Platform (RHOSP) 自定义选项。



重要

此示例文件仅供参考。您必须使用安装程序来获取 `install-config.yaml` 文件。

```
apiVersion: v1
baseDomain: example.com
clusterID: os-test
controlPlane:
  name: master
  platform: {}
  replicas: 3
compute:
- name: worker
  platform:
    openstack:
      type: ml.large
  replicas: 3
metadata:
  name: example
networking:
  clusterNetwork:
  - cidr: 10.128.0.0/14
    hostPrefix: 23
  machineNetwork:
  - cidr: 10.0.0.0/16
  serviceNetwork:
  - 172.30.0.0/16 1
  networkType: Kuryr
platform:
  openstack:
    cloud: mycloud
    externalNetwork: external
```

```

computeFlavor: m1.xlarge
lbFloatingIP: 128.0.0.1
trunkSupport: true 2
octaviaSupport: true 3
pullSecret: '{"auths": ...}'
sshKey: ssh-ed25519 AAAA...

```

- 1 Amphora Octavia 驱动程序为每个负载均衡器创建两个端口。因此，安装程序创建的服务子网是由 **serviceNetwork** 属性值指定的 CIDR 的两倍。要防止 IP 地址冲突，则需要更大的范围。
- 2 3 安装程序会自动发现 **trunkSupport** 和 **octaviaSupport**，因此无需设置它们。但是，如果您的环境不满足这两个要求，Kuryr SDN 将无法正常工作。需要使用中继来把 pod 连接到 RHOSP 网络，并且需要 Octavia 来创建 OpenShift Container Platform 服务。

1.2.10.8. Kuryr 端口池

Kuryr 端口池在待机时维护多个端口，用于创建 pod。

将端口保留在待机上可最大程度缩短 pod 创建时间。如果没有端口池，Kuryr 必须明确请求在创建或删除 pod 时创建或删除端口。

Kuryr 使用的 Neutron 端口是在绑定到命名空间的子网中创建的。这些 pod 端口也作为子端口添加到 OpenShift Container Platform 集群节点的主端口。

因为 Kuryr 将每个命名空间保留在单独的子网中，所以对于每个“命名空间-worker”对都会维护一个单独的端口池。

在安装集群前，您可以在 **cluster-network-03-config.yml** 清单文件中设置以下参数来配置端口池行为：

- **enablePortPoolsPrepopulation** 参数控制池预填充，它会强制 Kuryr 在创建时（如添加新主机或创建新命名空间时）将端口添加到池中。默认值为：**false**。
- **poolMinPorts** 参数是池中保留的最少可用端口的数量。默认值为：**1**。
- **poolMaxPorts** 参数是池中保留的最大可用端口数。如果值为 **0**，会禁用上限。这是默认的设置。
如果您的 OpenStack 端口配额较低，或者 pod 网络上的 IP 地址有限，请考虑设置此选项以确保删除不需要的端口。
- **poolBatchPorts** 参数定义一次可以创建的 Neutron 端口的最大数量。默认值为 **3**。

1.2.10.9. 在安装过程中调整 Kuryr 端口池

在安装过程中，您可以配置 Kuryr 如何管理 Red Hat OpenStack Platform (RHOSP) Neutron 端口，以控制 pod 创建的速度和效率。

先决条件

- 创建并修改 **install-config.yaml** 文件。

流程

1. 在命令行中创建清单文件：

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

- 1 对于 **<installation_directory>**，请指定含有集群的 **install-config.yaml** 文件的目录的名称。

2. 在 **<installation_directory>/manifests/** 目录下，创建一个名为 **cluster-network-03-config.yml** 的文件：

```
$ touch <installation_directory>/manifests/cluster-network-03-config.yml 1
```

- 1 对于 **<installation_directory>**，请指定包含集群的 **manifests/** 目录的目录名称。

创建该文件后，**manifests/** 目录中会包含多个网络配置文件，如下所示：

```
$ ls <installation_directory>/manifests/cluster-network-*
```

输出示例

```
cluster-network-01-crd.yml
cluster-network-02-config.yml
cluster-network-03-config.yml
```

3. 在编辑器中打开 **cluster-network-03-config.yml** 文件，并输入描述您想要的 Cluster Network Operator 配置的自定义资源(CR)：

```
$ oc edit networks.operator.openshift.io cluster
```

4. 编辑设置以满足您的要求。以下示例提供了以下文件：

```
apiVersion: operator.openshift.io/v1
kind: Network
metadata:
  name: cluster
spec:
  clusterNetwork:
  - cidr: 10.128.0.0/14
    hostPrefix: 23
  serviceNetwork:
  - 172.30.0.0/16
  defaultNetwork:
    type: Kuryr
    kuryrConfig:
      enablePortPoolsPrepopulation: false 1
      poolMinPorts: 1 2
      poolBatchPorts: 3 3
      poolMaxPorts: 5 4
      openstackServiceNetwork: 172.30.0.0/15 5
```

- 1 将 **enablePortPoolsPrepopulation** 的值设置为 **true** 以使 Kuryr 在创建命名空间或在集群中添加新节点后创建新 Neutron 端口。此设置引发 Neutron 端口配额，但可以缩短生成容器集所需的时间。默认值为：**false**。

- 2 如果池中的可用端口数量低于 **poolMinPorts** 的值，Kuryr 会为池创建新端口。默认值为 **1**。
- 3 **poolBatchPorts** 控制在可用端口数量低于 **poolMinPorts** 值时创建的新端口数量。默认值为 **3**。
- 4 如果池中的可用端口数量大于 **poolMaxPorts** 的值，Kuryr 会删除它们，直到数量与这个值匹配为止。将此值设置为 **0** 可禁用此上限，防止池缩小。默认值为 **0**。
- 5 **openStackServiceNetwork** 参数定义将 IP 地址分配到 RHOSP Octavia 的 LoadBalancer 的网络的 CIDR 范围。

如果此参数与 Amphora 驱动程序一起使用，则 Octavia 会为每个负载均衡器从这个网络获取两个 IP 地址：一个用于 OpenShift，另一个用于 VRRP 连接。由于这些 IP 地址分别由 OpenShift Container Platform 和 Neutron 管理，因此它们必须来自不同的池。因此，**openStackServiceNetwork** 的值必须至少是 **serviceNetwork** 值的两倍，**serviceNetwork** 的值必须与 **openStackServiceNetwork** 定义的范围完全重叠。

CNO 验证从此参数定义的范围获取的 VRRP IP 地址是否与 **serviceNetwork** 参数定义的范围不重叠。

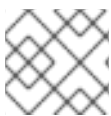
如果没有设置此参数，CNO 将使用 **serviceNetwork** 的扩展值，它是前缀大小值减 1。

5. 保存 **cluster-network-03-config.yml** 文件，再退出文本编辑器。
6. 可选：备份 **manifests/cluster-network-03-config.yml** 文件。安装程序在创建集群时删除 **manifests/** 目录。

1.2.11. 设置计算机器关联性

另外，您还可以在安装过程中为计算机器设置关联性策略。默认情况下，安装程序不会为计算机器选择关联性策略。

您还可以在安装后创建使用特定 RHOSP 服务器组的机器集。



注意

control plane 机器使用 **soft-anti-affinity** 策略创建。

提示

您可以在 RHOSP 文档中了解更多有关 [RHOSP 实例调度和放置](#) 的信息。

先决条件

- 创建 **install-config.yaml** 文件并完成对其所做的任何修改。

流程

1. 使用 RHOSP 命令行界面，为您的计算机器创建服务器组。例如：

```
$ openstack \
  --os-compute-api-version=2.15 \
  server group create \
```

```
--policy anti-affinity \
my-openshift-worker-group
```

如需更多信息，请参阅[服务器组 create 命令文档](#)。

2. 进入包含安装程序的目录并创建清单：

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

其中：

installation_directory

指定包含集群的 `install-config.yaml` 文件的目录名称。

3. 打开 `manifests/99_openshift-cluster-api_worker-machineset-0.yaml`，这是 `MachineSet` 定义文件。
4. 将属性 `serverGroupID` 添加到 `spec.template.spec.providerSpec.value` 属性下的定义中。例如：

```
apiVersion: machine.openshift.io/v1beta1
kind: MachineSet
metadata:
  labels:
    machine.openshift.io/cluster-api-cluster: <infrastructure_ID>
    machine.openshift.io/cluster-api-machine-role: <node_role>
    machine.openshift.io/cluster-api-machine-type: <node_role>
  name: <infrastructure_ID>-<node_role>
  namespace: openshift-machine-api
spec:
  replicas: <number_of_replicas>
  selector:
    matchLabels:
      machine.openshift.io/cluster-api-cluster: <infrastructure_ID>
      machine.openshift.io/cluster-api-machineset: <infrastructure_ID>-<node_role>
  template:
    metadata:
      labels:
        machine.openshift.io/cluster-api-cluster: <infrastructure_ID>
        machine.openshift.io/cluster-api-machine-role: <node_role>
        machine.openshift.io/cluster-api-machine-type: <node_role>
        machine.openshift.io/cluster-api-machineset: <infrastructure_ID>-<node_role>
    spec:
      providerSpec:
        value:
          apiVersion: openstackproviderconfig.openshift.io/v1alpha1
          cloudName: openstack
          cloudsSecret:
            name: openstack-cloud-credentials
            namespace: openshift-machine-api
          flavor: <nova_flavor>
          image: <glance_image_name_or_location>
          serverGroupID: aaaaaaa-bbbb-cccc-dddd-eeeeeeeeeeee 1
          kind: OpenstackProviderSpec
          networks:
```

```

- filter: {}
  subnets:
  - filter:
    name: <subnet_name>
    tags: openshiftClusterID=<infrastructure_ID>
  securityGroups:
  - filter: {}
    name: <infrastructure_ID>-<node_role>
  serverMetadata:
    Name: <infrastructure_ID>-<node_role>
    openshiftClusterID: <infrastructure_ID>
  tags:
  - openshiftClusterID=<infrastructure_ID>
  trunk: true
  userDataSecret:
    name: <node_role>-user-data
  availabilityZone: <optional_openstack_availability_zone>

```

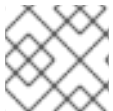
❶ 在此处添加服务器组的 UUID。

5. 可选：备份 **manifests/99_openshift-cluster-api_worker-machineset-0.yaml** 文件。创建集群时，安装程序会删除 **manifests/** 目录。

安装集群时，安装程序将使用您修改的 **MachineSet** 定义在 RHOSP 服务器组中创建计算机。

1.2.12. 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。

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> ❶

```

❶ 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。

1.2.13. 启用对环境的访问

在部署时，所有 OpenShift Container Platform 机器都是在 Red Hat OpenStack Platform (RHOSP) 租户网络中创建的。因此，大多数 RHOSP 部署中都无法直接访问它们。

您可以在安装过程中使用浮动 IP 地址（FIP）来配置 OpenShift Container Platform API 和应用程序访问。您也可以在没有配置 FIP 的情况下完成安装，但安装程序不会配置一种从外部访问 API 或应用程序的方法。

1.2.13.1. 启用通过浮动 IP 地址进行访问

创建浮动 IP（FIP）地址，用于从外部访问 OpenShift Container Platform API 和集群应用程序。

流程

1. 使用 Red Hat OpenStack Platform (RHOSP) CLI，创建 API FIP：

```
$ openstack floating ip create --description "API <cluster_name>.<base_domain>"
<external_network>
```

2. 使用 Red Hat OpenStack Platform (RHOSP) CLI，创建应用程序或 Ingress，FIP：

```
$ openstack floating ip create --description "Ingress <cluster_name>.<base_domain>"
<external_network>
```

3. 向用于 API 和 Ingress FIP 的 DNS 服务器添加符合这些模式的记录：

```
api.<cluster_name>.<base_domain>. IN A <API_FIP>
*.apps.<cluster_name>.<base_domain>. IN A <apps_FIP>
```

注意

如果您不控制 DNS 服务器，您可以通过将集群域名（如以下内容）添加到 `/etc/hosts` 文件中来访问集群：

- `<api_floating_ip> api.<cluster_name>.<base_domain>`
- `<application_floating_ip> grafana-openshift-monitoring.apps.<cluster_name>.<base_domain>`
- `<application_floating_ip> prometheus-k8s-openshift-monitoring.apps.<cluster_name>.<base_domain>`
- `<application_floating_ip> oauth-openshift.apps.<cluster_name>.<base_domain>`
- `<application_floating_ip> console-openshift-console.apps.<cluster_name>.<base_domain>`
- `application_floating_ip integrate-oauth-server-openshift-authentication.apps.<cluster_name>.<base_domain>`

`/etc/hosts` 文件中的集群域名授予对本地集群的 Web 控制台和监控界面的访问权限。您还可以使用 `kubectl` 或 `oc`。您可以使用指向 `<application_floating_ip>` 的额外条目来访问用户应用程序。此操作使 API 和应用程序可供您访问，不适用于生产部署，但允许对开发和测试进行安装。

4. 将 FIP 添加到 `install-config.yaml` 文件，将其作为以下参数的值：

- `platform.openstack.ingressFloatingIP`
- `platform.openstack.lbFloatingIP`

如果使用这些值，还必须在 `install-config.yaml` 文件中输入一个外部网络作为 `platform.openstack.externalNetwork` 参数的值。

提示

您可以通过分配浮动 IP 地址并更新防火墙配置，使 OpenShift Container Platform 资源在集群之外可用。

1.2.13.2. 完成没有浮动 IP 地址的安装

您可以在不提供浮动 IP 地址的情况下在 Red Hat OpenStack Platform (RHOSP) 上安装 OpenShift Container Platform。

在 `install-config.yaml` 文件中，不要定义以下参数：

- `platform.openstack.ingressFloatingIP`
- `platform.openstack.lbFloatingIP`

如果您无法提供外部网络，也可以将 `platform.openstack.externalNetwork` 留空。如果没有为 `platform.openstack.externalNetwork` 提供值，则不会为您创建路由器。如果没有额外的操作，安装程序将无法从 Glance 检索镜像。您必须自行配置外部连接。

如果在因为缺少浮动 IP 地址或名称解析而无法访问集群 API 的系统中运行安装程序时，安装会失败。要防止安装失败，可以使用代理网络或者从与您的机器位于同一网络的系统中运行安装程序。



注意

您可以通过为 API 和 Ingress 端口创建 DNS 记录来启用名称解析。例如：

```
api.<cluster_name>.<base_domain>. IN A <api_port_IP>
*.apps.<cluster_name>.<base_domain>. IN A <ingress_port_IP>
```

如果您不控制 DNS 服务器，可以改为将记录添加到 `/etc/hosts` 文件中。此操作使 API 可供您自己访问，不适用于生产部署。这可用于进行开发和测试的安装。

1.2.14. 部署集群

您可以在兼容云平台中安装 OpenShift Container Platform。



重要

安装程序的 `create cluster` 命令只能在初始安装过程中运行一次。

先决条件

- 获取 OpenShift Container Platform 安装程序以及集群的 pull secret。

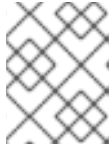
流程

1. 更改为包含安装程序的目录并初始化集群部署：

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

1 对于 `<installation_directory>`，请指定自定义 `./install-config.yaml` 文件的位置。

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



注意

如果您在主机上配置的云供应商帐户没有足够的权限来部署集群，安装过程将会停止，并且显示缺少的权限。

集群部署完成后，终端会显示访问集群的信息，包括指向其 Web 控制台的链接和 **kubeadmin** 用户的凭证。

输出示例

```

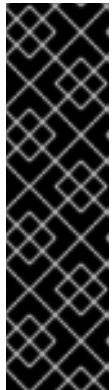
...
INFO Install complete!
INFO To access the cluster as the system:admin user when using 'oc', run 'export
KUBECONFIG=/home/myuser/install_dir/auth/kubeconfig'
INFO Access the OpenShift web-console here: https://console-openshift-
console.apps.mycluster.example.com
INFO Login to the console with user: "kubeadmin", and password: "4vYBz-Ee6gm-ymBZj-
Wt5AL"
INFO Time elapsed: 36m22s

```



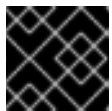
注意

当安装成功时，集群访问和凭证信息还会输出到 **<installation_directory>/openshift_install.log**。



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-bootstrap** 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。



重要

您不得删除安装程序或安装程序所创建的文件。需要这两者才能删除集群。

1.2.15. 验证集群状态

您可以在安装过程中或安装后验证 OpenShift Container Platform 集群的状态：

流程

1. 在集群环境中，导出管理员的 kubeconfig 文件：

```
$ 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。

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。

2. 查看部署后创建的 control plane 和计算机器：

```
$ oc get nodes
```

3. 查看集群的版本：

```
$ oc get clusterversion
```

4. 查看 Operator 的状态：

```
$ oc get clusteroperator
```

5. 查看集群中的所有正在运行的 pod:

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

1.2.16. 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
```

其他资源

- 如需有关访问和了解 OpenShift Container Platform Web 控制台的更多信息，请参阅[访问 Web 控制台](#)。

1.2.17. 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](#)。

1.2.18. 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](#)。
- 如果您需要启用对节点端口的外部访问，[请使用节点端口配置集群流量](#)。
- 如果您没有将 RHOSP 配置为使用浮动 IP 地址接受应用程序流量，[使用浮动 IP 地址配置 RHOSP 访问](#)。

1.3. 在您自己的基础架构的 OPENSTACK 上安装集群

在 OpenShift Container Platform 版本 4.6 中，您可以在运行于用户自备的基础架构上的 Red Hat OpenStack Platform (RHOSP) 上安装集群。

通过利用您自己的基础架构，您可以将集群与现有的基础架构进行集成。和安装程序自备的安装方式相比，这个过程需要用户进行更多操作，因为您必须创建所有 RHOSP 资源，如 Nova 服务器、Neutron 端口和安全组。红帽提供了 Ansible playbook 来帮助您完成部署过程。

1.3.1. 先决条件

- 查看有关 [OpenShift Container Platform 安装和更新](#) 流程的详细信息。
 - 在 *Available platforms* 部分验证 OpenShift Container Platform 4.6 是否与您的 RHOSP 版本兼容。您还可以查看 [OpenShift Container Platform 在 RHOSP 中的支持](#) 来比较不同版本的平台支持。
- 验证您的网络配置不依赖于供应商网络。不支持提供商网络。
- 具有要安装 OpenShift Container Platform 的 RHOSP 帐户
- 在您运行安装程序的机器中，有：
 - 用来保存在安装过程中创建的文件的一个单一目录
 - Python 3

1.3.2. 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。

1.3.3. 在 RHOSP 上安装 OpenShift Container Platform 的资源指南

您的 Red Hat OpenStack Platform（RHOSP）配额需要满足以下条件才支持 OpenShift Container Platform 安装：

表 1.13. RHOSP 上默认 OpenShift Container Platform 集群的建议资源

资源	值
浮动 IP 地址	3
端口	15
路由器	1
子网	1
RAM	112 GB
vCPUs	28
卷存储	275 GB
实例	7
安全组	3
安全组规则	60

集群或许能使用少于推荐数量的资源来运作，但其性能无法保证。



重要

如果 RHOSP 对象存储 (Swift) 可用, 并由具有 **swiftoperator** 角色的用户帐户执行, 它会作为 OpenShift Container Platform 镜像 registry 的默认后端。在这种情况下, 卷存储需要有 175GB。根据镜像 registry 的大小, Swift 空间要求会有所不同。



注意

默认情况下, 您的安全组和安全组规则配额可能较低。如果遇到问题, 请以 admin 的身份运行 **openstack quota set --secgroups 3 --secgroup-rules 60 <project>** 来提高配额。

OpenShift Container Platform 部署由 control plane 机器、计算机器和 bootstrap 机器组成。

1.3.3.1. control plane 机器

默认情况下, OpenShift Container Platform 安装过程会创建三台 control plane 机器。

每台机器都需要：

- 来自 RHOSP 配额的实例
- 来自 RHOSP 配额的端口
- 至少有 16 GB 内存、4 个 vCPU 和 100 GB 存储空间类别

1.3.3.2. 计算机器

默认情况下, OpenShift Container Platform 安装过程会创建三台计算机器。

每台机器都需要：

- 来自 RHOSP 配额的实例
- 来自 RHOSP 配额的端口
- 至少有 8 GB 内存、2 个 vCPU 和 100 GB 存储空间类别

提示

计算机器托管您在 OpenShift Container Platform 上运行的应用程序；运行数量应尽可能多。

1.3.3.3. bootstrap 机器

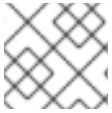
在安装时, 会临时置备 bootstrap 机器来支持 control plane。生产控制平面就绪后, bootstrap 机器会被取消置备。

bootstrap 机器需要：

- 来自 RHOSP 配额的实例
- 来自 RHOSP 配额的端口
- 至少有 16 GB 内存、4 个 vCPU 和 100 GB 存储空间类别

1.3.4. 下载 playbook 的依赖项

简化用户置备基础架构安装过程的 Ansible playbook 需要几个 Python 模块。在您要运行安装程序的机器上添加模块的仓库，然后下载它们。



注意

这些说明假设您使用 Red Hat Enterprise Linux (RHEL) 8。

先决条件

- Python 3 已安装在您的机器上。

流程

1. 在命令行中添加软件仓库：

- a. 使用 Red Hat Subscription Manager 注册：

```
$ sudo subscription-manager register # If not done already
```

- b. 获取最新的订阅数据：

```
$ sudo subscription-manager attach --pool=$YOUR_POOLID # If not done already
```

- c. 禁用当前的软件仓库：

```
$ sudo subscription-manager repos --disable=* # If not done already
```

- d. 添加所需的软件仓库：

```
$ sudo subscription-manager repos \
  --enable=rhel-8-for-x86_64-baseos-rpms \
  --enable=openstack-16-tools-for-rhel-8-x86_64-rpms \
  --enable=ansible-2.9-for-rhel-8-x86_64-rpms \
  --enable=rhel-8-for-x86_64-appstream-rpms
```

2. 安装模块：

```
$ sudo yum install python3-openstackclient ansible python3-openstacksdk python3-netaddr
```

3. 确保 **python** 命令指向 **python3**：

```
$ sudo alternatives --set python /usr/bin/python3
```

1.3.5. 下载安装 playbook

下载 Ansible playbook，可用于在您自己的 Red Hat OpenStack Platform (RHOSP) 基础架构上安装 OpenShift Container Platform。

先决条件

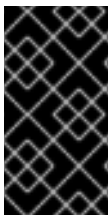
- curl 命令行工具可在您的机器上找到。

流程

- 要将 playbook 下载到您的工作目录中，请从命令行运行以下脚本：

```
$ xargs -n 1 curl -O <<< '
  https://raw.githubusercontent.com/openshift/installer/release-
4.6/upi/openshift/bootstrap.yaml
  https://raw.githubusercontent.com/openshift/installer/release-
4.6/upi/openshift/common.yaml
  https://raw.githubusercontent.com/openshift/installer/release-
4.6/upi/openshift/compute-nodes.yaml
  https://raw.githubusercontent.com/openshift/installer/release-4.6/upi/openshift/control-
plane.yaml
  https://raw.githubusercontent.com/openshift/installer/release-
4.6/upi/openshift/inventory.yaml
  https://raw.githubusercontent.com/openshift/installer/release-
4.6/upi/openshift/network.yaml
  https://raw.githubusercontent.com/openshift/installer/release-4.6/upi/openshift/security-
groups.yaml
  https://raw.githubusercontent.com/openshift/installer/release-4.6/upi/openshift/down-
bootstrap.yaml
  https://raw.githubusercontent.com/openshift/installer/release-4.6/upi/openshift/down-
compute-nodes.yaml
  https://raw.githubusercontent.com/openshift/installer/release-4.6/upi/openshift/down-
control-plane.yaml
  https://raw.githubusercontent.com/openshift/installer/release-4.6/upi/openshift/down-
load-balancers.yaml
  https://raw.githubusercontent.com/openshift/installer/release-4.6/upi/openshift/down-
network.yaml
  https://raw.githubusercontent.com/openshift/installer/release-4.6/upi/openshift/down-
security-groups.yaml
  https://raw.githubusercontent.com/openshift/installer/release-4.6/upi/openshift/down-
containers.yaml'
```

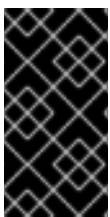
playbook 下载到您的机器中。



重要

在安装过程中，您可以修改 playbook 来配置部署。

在集群生命周期中保留所有 playbook。您必须具有 playbook，才能从 RHOSP 中删除 OpenShift Container Platform 集群。



重要

您在 **bootstrap.yaml**、**compute-nodes.yaml**、**control-plane.yaml**、**network.yaml** 和 **security-groups.yaml** 文件中进行的任何改变都需要与带有 **down-** 前缀的对应的 playbook 相匹配。例如，对 **bootstrap.yaml** 文件的编辑也必须反映在 **down-bootstrap.yaml** 文件中。如果没有编辑这两个文件，则支持的删除集群过程将失败。

1.3.6. 获取安装程序

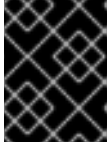
在安装 OpenShift Container Platform 之前，将安装文件下载到本地计算机上。

先决条件

- 运行 Linux 或 macOS 的计算机，本地磁盘空间为 500 MB

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。

1.3.7. 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。

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。



注意

您必须使用一个本地密钥，而不要使用在特定平台上配置的密钥，如 [AWS 密钥对](#)。

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

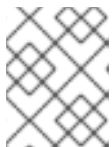
如果您计划在 `x86_64` 架构中安装使用 FIPS 验证的/Modules in Process 加密库的 OpenShift Container Platform 集群，不要创建使用 `ed25519` 算法的密钥。反之，创建一个使用 `rsa` 或 `ecdsa` 算法的密钥。

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。

1.3.8. 创建 Red Hat Enterprise Linux CoreOS (RHCOS) 镜像

OpenShift Container Platform 安装程序要求 Red Hat OpenStack Platform (RHOSP) 集群中有 Red Hat Enterprise Linux CoreOS (RHCOS) 镜像。检索最新的 RHCOS 镜像，然后使用 RHOSP CLI 上传该镜像。

先决条件

- 已安装了 RHOSP CLI。

流程

1. 登录到红帽客户门户网站的[产品下载页](#)。
2. 在 **Version** 下，为 Red Hat Enterprise Linux (RHEL) 8 选择 OpenShift Container Platform 4.6 的最新发行版本。



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。如果可用，请使用与 OpenShift Container Platform 版本匹配的镜像版本。

3. 下载 *Red Hat Enterprise Linux CoreOS (RHCOS) - OpenStack Image (QCOW)* 。
4. 解压镜像。



注意

您必须解压 RHOSP 镜像，然后集群才能使用它。下载的文件名可能不包含压缩扩展名，如 **.gz** 或 **.tgz**。要找出是否或者如何压缩文件，请在命令行中输入：

```
$ file <name_of_downloaded_file>
```

5. 从您下载的镜像，使用 RHOSP CLI 在集群中创建名为 **rhcos** 的镜像：

```
$ openstack image create --container-format=bare --disk-format=qcow2 --file rhcos-
${RHCOS_VERSION}-openstack.qcow2 rhcos
```



重要

根据您的 RHOSP 环境，可能需要使用 **.raw** 或 **.qcow2** 格式下载镜像。如果使用 Ceph，则必须使用 **.raw** 格式。



警告

如果安装程序发现多个同名的镜像，它会随机选择其中之一。为避免这种行为，请在 RHOSP 中为资源创建唯一名称。

将镜像上传到 RHOSP 后，就可以被安装程序使用。

1.3.9. 验证外部网络访问

OpenShift Container Platform 安装进程需要外部网络访问权限。您必须为其提供外部网络值，否则部署会失败。在运行安装进程前，请验证 Red Hat OpenStack Platform (RHOSP) 中是否存在具有外部路由器类型的网络。

先决条件

- 将 OpenStack 联网服务配置为使用 DHCP 代理转发实例 DNS 查询

流程

1. 使用 RHOSP CLI 验证“外部”网络的名称和 ID :

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

输出示例

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

网络列表中会显示具有外部路由器类型的网络。如果最少有一个没有，请参阅 [创建默认浮动 IP 网络](#)和 [创建默认供应商网络](#)。



注意

如果启用了 Neutron 中继服务插件，则默认创建中继端口。如需更多信息，请参阅 [Neutron 中继端口](#)。

1.3.10. 启用对环境的访问

在部署时，所有 OpenShift Container Platform 机器都是在 Red Hat OpenStack Platform (RHOSP) 租户网络中创建的。因此，大多数 RHOSP 部署中都无法直接访问它们。

您可以在安装过程中使用浮动 IP 地址 (FIP) 来配置 OpenShift Container Platform API 和应用程序访问。您也可以在没有配置 FIP 的情况下完成安装，但安装程序不会配置一种从外部访问 API 或应用程序的方法。

1.3.10.1. 启用通过浮动 IP 地址进行访问

创建浮动 IP(FIP)地址，用于从外部访问 OpenShift Container Platform API、集群应用程序和 bootstrap 过程。

流程

1. 使用 Red Hat OpenStack Platform (RHOSP) CLI，创建 API FIP :

```
$ openstack floating ip create --description "API <cluster_name>.<base_domain>"
<external_network>
```

2. 使用 Red Hat OpenStack Platform (RHOSP) CLI，创建应用程序或 Ingress，FIP :

```
$ openstack floating ip create --description "Ingress <cluster_name>.<base_domain>"
<external_network>
```

3. 使用 Red Hat OpenStack Platform (RHOSP) CLI 创建 bootstrap FIP:

```
$ openstack floating ip create --description "bootstrap machine" <external_network>
```

4. 向用于 API 和 Ingress FIP 的 DNS 服务器添加符合这些模式的记录：

```
api.<cluster_name>.<base_domain>. IN A <API_FIP>
*.apps.<cluster_name>.<base_domain>. IN A <apps_FIP>
```

注意

如果您不控制 DNS 服务器，您可以通过将集群域名（如以下内容）添加到 `/etc/hosts` 文件中来访问集群：

- `<api_floating_ip> api.<cluster_name>.<base_domain>`
- `<application_floating_ip> grafana-openshift-monitoring.apps.<cluster_name>.<base_domain>`
- `<application_floating_ip> prometheus-k8s-openshift-monitoring.apps.<cluster_name>.<base_domain>`
- `<application_floating_ip> oauth-openshift.apps.<cluster_name>.<base_domain>`
- `<application_floating_ip> console-openshift-console.apps.<cluster_name>.<base_domain>`
- `application_floating_ip integrate-oauth-server-openshift-authentication.apps.<cluster_name>.<base_domain>`

`/etc/hosts` 文件中的集群域名授予对本地集群的 Web 控制台和监控界面的访问权限。您还可以使用 `kubectl` 或 `oc`。您可以使用指向 `<application_floating_ip>` 的额外条目来访问用户应用程序。此操作使 API 和应用程序可供您访问，不适用于生产部署，但允许对开发和测试进行安装。

5. 将 FIP 添加到 `inventory.yaml` 文件，作为以下变量的值：

- `os_api_fip`
- `os_bootstrap_fip`
- `os_ingress_fip`

如果使用这些值，还必须在 `inventory.yaml` 文件中输入一个外部网络作为 `os_external_network` 变量的值。

提示

您可以通过分配浮动 IP 地址并更新防火墙配置，使 OpenShift Container Platform 资源在集群之外可用。

1.3.10.2. 完成没有浮动 IP 地址的安装

您可以在不提供浮动 IP 地址的情况下在 Red Hat OpenStack Platform (RHOSP) 上安装 OpenShift Container Platform。

在 `inventory.yaml` 文件中，不要定义以下变量：

- `os_api_fip`
- `os_bootstrap_fip`
- `os_ingress_fip`

如果无法提供外部网络，也可以将 `os_external_network` 留空。如果没有为 `os_external_network` 提供值，则不会为您创建路由器。如果没有额外的操作，安装程序将无法从 Glance 检索镜像。之后在安装过程中，当您创建网络资源时，必须自行配置外部连接。

如果您使用 `wait-for` 命令从因为缺少浮动 IP 地址或名称解析而无法访问集群 API 的系统中运行安装程序时，安装会失败。要防止安装失败，可以使用代理网络或者从与您的机器位于同一网络的系统中运行安装程序。



注意

您可以通过为 API 和 Ingress 端口创建 DNS 记录来启用名称解析。例如：

```
api.<cluster_name>.<base_domain>. IN A <api_port_IP>
*.apps.<cluster_name>.<base_domain>. IN A <ingress_port_IP>
```

如果您不控制 DNS 服务器，可以改为将记录添加到 `/etc/hosts` 文件中。此操作使 API 可供您自己访问，不适合于生产部署。这可用于进行开发和测试的安装。

1.3.11. 为安装程序定义参数

OpenShift Container Platform 安装程序依赖于一个名为 `clouds.yaml` 的文件。该文件描述了 Red Hat OpenStack Platform (RHOSP) 配置参数，包括项目名称、登录信息和授权服务 URL。

流程

1. 创建 `clouds.yaml` 文件：

- 如果您的 RHOSP 发行版包含 Horizon web UI，请在该 UI 中生成 `clouds.yaml` 文件。



重要

请记住在 `auth` 字段中添加密码。您也可以把 secret 保存在 `clouds.yaml` 以外的一个独立的文件中。

- 如果您的 RHOSP 发行版不包含 Horizon Web UI，或者您不想使用 Horizon，请自行创建该文件。如需有关 `clouds.yaml` 的详细信息，请参阅 RHOSP 文档中的 [配置文件](#)。

```
clouds:
  shiftstack:
    auth:
      auth_url: http://10.10.14.42:5000/v3
      project_name: shiftstack
      username: shiftstack_user
      password: XXX
      user_domain_name: Default
      project_domain_name: Default
```

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

2. 如果您的 RHOSP 安装使用自签名证书颁发机构 (CA) 证书进行端点身份验证：

- a. 将 CA 文件复制到您的机器中。
- b. 将机器添加到证书颁发机构信任捆绑包中：

```
$ sudo cp ca.crt.pem /etc/pki/ca-trust/source/anchors/
```

- c. 更新信任捆绑包：

```
$ sudo update-ca-trust extract
```

- d. 将 **cacerts** 键添加到 **clouds.yaml** 文件。该值必须是到 CA 证书的绝对路径，则其可以被非根用户访问：

```
clouds:
  shiftstack:
    ...
  cacert: "/etc/pki/ca-trust/source/anchors/ca.crt.pem"
```

提示

使用自定义 CA 证书运行安装程序后，您可以通过编辑 **cloud-provider-config** keymap 中的 **ca-cert.pem** 键的值来更新证书。在命令行中运行：

```
$ oc edit configmap -n openshift-config cloud-provider-config
```

3. 将 **clouds.yaml** 文件放在以下位置之一：

- a. **OS_CLIENT_CONFIG_FILE** 环境变量的值
- b. 当前目录
- c. 特定于 Unix 的用户配置目录，如 **~/.config/openshift/clouds.yaml**
- d. 特定于 Unix 的站点配置目录，如 **/etc/openshift/clouds.yaml**
安装程序会按照以上顺序搜索 **clouds.yaml**。

1.3.12. 创建安装配置文件

您可以自定义在 Red Hat OpenStack Platform (RHOSP) 上安装的 OpenShift Container Platform 集群。

先决条件

- 获取 OpenShift Container Platform 安装程序以及集群的 pull secret。

流程

1. 创建 `install-config.yaml` 文件。
 - a. 更改到包含安装程序的目录，再运行以下命令：

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

- 1** 对于 `<installation_directory>`，请指定用于保存安装程序所创建的文件目录名称。



重要

指定一个空目录。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。

- b. 在提示符处，提供您的云的配置详情：
 - i. 可选：选择用来访问集群机器的 SSH 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。

- ii. 选择 `openstack` 作为目标平台。
 - iii. 指定用于安装集群的 Red Hat OpenStack Platform (RHOSP) 外部网络名称。
 - iv. 指定用于从外部访问 OpenShift API 的浮动 IP 地址。
 - v. 指定至少有 16 GB RAM 用于 control plane 和计算节点的 RHOSP 类别。
 - vi. 选择集群要部署到的基域。所有 DNS 记录都将是这个基域的子域，并包含集群名称。
 - vii. 为集群输入一个名称。名称不能多于 14 个字符。
 - viii. 粘贴 [Red Hat OpenShift Cluster Manager 中的 pull secret](#)。
2. 修改 `install-config.yaml` 文件。您可以在 [安装配置参数](#) 部分中找到有关可用参数的更多信息。
 3. 备份 `install-config.yaml` 文件，以便用于安装多个集群。



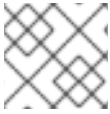
重要

`install-config.yaml` 文件会在安装过程中消耗掉。如果要重复使用此文件，必须现在备份。

现在，文件 `install-config.yaml` 位于您指定的目录中。

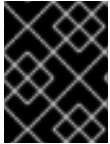
1.3.13. 安装配置参数

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。

1.3.13.1. 所需的配置参数

下表描述了所需的安装配置参数：

表 1.14. 所需的参数

参数	描述	值
<code>apiVersion</code>	<code>install-config.yaml</code> 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
<code>baseDomain</code>	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 <code>baseDomain</code> 和 <code>metadata.name</code> 参数值的组合，其格式为 <code><metadata.name>.<baseDomain></code> 。	完全限定域名或子域名，如 example.com 。
<code>metadata</code>	Kubernetes 资源 ObjectMeta ，其中只消耗 <code>name</code> 参数。	对象
<code>metadata.name</code>	集群的名称。集群的 DNS 记录是 <code>{{.metadata.name}}</code> 。 <code>{{.baseDomain}}</code> 的子域。	小写字母、连字符(-)和句点(.)的字符串，如 dev 。该字符串长度必须为 14 个字符或更少。
<code>platform</code>	执行安装的具体平台配置： aws 、 baremetal 、 azure 、 openstack 、 ovirt 、 vsphere 。有关 <code>platform.<platform></code> 参数的额外信息，请参考下表来了解您的具体平台。	对象


参数	描述	值
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>

1.3.13.2. 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 1.15. 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： <pre>networking: clusterNetwork: - cidr: 10.128.0.0/14 hostPrefix: 23</pre>
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。

参数	描述	值
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： <pre>networking: serviceNetwork: - 172.30.0.0/16</pre>
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。	一个对象数组。例如： <pre>networking: machineNetwork: - cidr: 10.0.0.0/16</pre>
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。

1.3.13.3. 可选配置参数

下表描述了可选安装配置参数：

表 1.16. 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。

参数	描述	值
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	<p>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。</p> <div style="display: flex; align-items: center;">  <p>重要</p> </div> <p>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。</p>	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	<p>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。</p> <div style="display: flex; align-items: center;">  <p>重要</p> </div> <p>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。</p>	Enabled 或 Disabled

参数	描述	值
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	<p>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。</p> <div style="display: flex; align-items: flex-start;"> <div style="width: 40px; height: 40px; background: repeating-linear-gradient(45deg, transparent, transparent 2px, #ccc 2px, #ccc 4px); margin-right: 10px;"></div> <div> <p>注意</p> <p>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 <i>Red Hat Operator 参考指南</i> 内容中的 <i>Cloud Credential Operator</i> 条目。</p> </div> </div>	Mint、Passthrough、Manual 或空字符串("")。
fips	<p>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。</p> <div style="display: flex; align-items: flex-start;"> <div style="width: 40px; height: 40px; background: repeating-linear-gradient(45deg, transparent, transparent 2px, #000 2px, #000 4px); margin-right: 10px;"></div> <div> <p>重要</p> <p>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。</p> </div> </div> <div style="display: flex; align-items: flex-start; margin-top: 10px;"> <div style="width: 40px; height: 40px; background: repeating-linear-gradient(45deg, transparent, transparent 2px, #ccc 2px, #ccc 4px); margin-right: 10px;"></div> <div> <p>注意</p> <p>如果使用 Azure File 存储，则无法启用 FIPS 模式。</p> </div> </div>	false 或 true

参数	描述	值
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。	<p>Internal 或 External。默认值为 External。</p> <p>在非云平台上不支持将此字段设置为 Internal。</p> <div style="display: flex; align-items: flex-start;"> <div style="width: 60px; height: 60px; background: repeating-linear-gradient(45deg, transparent, transparent 2px, black 2px, black 4px); margin-right: 10px;"></div> <div> <p>重要</p> <p>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。</p> </div> </div>
sshKey	<p>用于验证集群机器访问的 SSH 密钥或密钥。</p> <div style="display: flex; align-items: flex-start;"> <div style="width: 60px; height: 60px; background: repeating-linear-gradient(-45deg, transparent, transparent 2px, black 2px, black 4px); margin-right: 10px;"></div> <div> <p>注意</p> <p>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。</p> </div> </div>	<p>一个或多个密钥。例如：</p> <pre>sshKey: <key1> <key2> <key3></pre>

1.3.13.4. 其他 Red Hat OpenStack Platform (RHOSP) 配置参数

下表描述了其他 RHOSP 配置参数：

表 1.17. 其他 RHOSP 参数

参数	描述	值
<code>compute.platform.openstack.rootVolume.size</code>	对于计算机器，以 GB 为单位表示的根卷大小。如果您不设置这个值，机器将使用临时存储。	整数，如 30 。
<code>compute.platform.openstack.rootVolume.type</code>	对于计算机器，根卷的类型。	字符串，如 performance 。
<code>controlPlane.platform.openstack.rootVolume.size</code>	对于 control plane 机器，以 GB 为单位表示的根卷大小。如果您不设置这个值，机器将使用临时存储。	整数，如 30 。
<code>controlPlane.platform.openstack.rootVolume.type</code>	对于 control plane 机器，根卷的类型。	字符串，如 performance 。
<code>platform.openstack.cloud</code>	要使用的 RHOSP 云的名称，来自于 <code>clouds.yaml</code> 文件中的云列表。	字符串，如 MyCloud 。
<code>platform.openstack.externalNetwork</code>	用于安装的 RHOSP 外部网络名称。	字符串，如 external 。
<code>platform.openstack.computeFlavor</code>	用于 control plane 和计算机器的 RHOSP 类别。	字符串，如 m1.xlarge 。

1.3.13.5. 可选 RHOSP 配置参数

下表描述了可选 RHOSP 配置参数：

表 1.18. 可选的 RHOSP 参数

参数	描述	值
<code>compute.platform.openstack.additionalNetworkIDs</code>	与计算机器关联的其他网络。不能为额外网络创建允许的地址对。	一个或多个 UUID 列表作为字符串。例如： fa806b2f-ac49-4bce-b9db-124bc64209bf 。
<code>compute.platform.openstack.additionalSecurityGroupIDs</code>	与计算机器关联的其他安全组。	一个或多个 UUID 列表作为字符串。例如： 7ee219f3-d2e9-48a1-96c2-e7429f1b0da7 。

参数	描述	值
compute.platform.openstack.zones	<p>RHOSP Compute (Nova) 可用区 (AZ) 在其中安装机器。如果没有设置此参数, 安装程序会依赖于配置了 RHOSP 管理员的 Nova 的默认设置。</p> <p>在使用 Kuryr 的集群上, RHOSP Octavia 不支持可用域。负载均衡器, 如果您使用 Amphora 供应商驱动程序, 则依赖 Amphora 虚拟机的 OpenShift Container Platform 服务不会根据此属性的值创建。</p>	字符串列表。例如: ["zone-1", "zone-2"]。
controlPlane.platform.openstack.additionalNetworkIDs	与 control plane 机器关联的额外网络。不能为额外网络创建允许的地址对。	一个或多个 UUID 列表作为字符串。例如: fa806b2f-ac49-4bce-b9db-124bc64209bf 。
controlPlane.platform.openstack.additionalSecurityGroupIDs	与 control plane 机器关联的其他安全组。	一个或多个 UUID 列表作为字符串。例如: 7ee219f3-d2e9-48a1-96c2-e7429f1b0da7 。
controlPlane.platform.openstack.zones	<p>RHOSP Compute (Nova) 可用区 (AZ) 在其中安装机器。如果没有设置此参数, 安装程序会依赖于配置了 RHOSP 管理员的 Nova 的默认设置。</p> <p>在使用 Kuryr 的集群上, RHOSP Octavia 不支持可用域。负载均衡器, 如果您使用 Amphora 供应商驱动程序, 则依赖 Amphora 虚拟机的 OpenShift Container Platform 服务不会根据此属性的值创建。</p>	字符串列表。例如: ["zone-1", "zone-2"]。
platform.openstack.clusterOSImage	<p>安装程序从中下载 RHCOS 镜像的位置。</p> <p>您必须设置此参数以便在受限网络中执行安装。</p>	<p>HTTP 或 HTTPS URL, 可选使用 SHA-256 checksum。</p> <p>例如: http://mirror.example.com/images/rhcos-43.81.201912131630.0-openstack.x86_64.qcow2.gz?sha256=ffebbd68e8a1f2a245ca19522c16c86f67f9ac8e4e0c1f0a812b068b16f7265d。该值也可以是现有 Glance 镜像的名称, 如 my-rhcos。</p>

参数	描述	值
platform.openstack.defaultMachinePlatform	默认机器池平台配置。	<pre>{ "type": "ml.large", "rootVolume": { "size": 30, "type": "performance" } }</pre>
platform.openstack.ingressFloatingIP	与 Ingress 端口关联的现有浮动 IP 地址。要使用此属性，还必须定义 platform.openstack.externalNetwork 属性。	IP 地址，如 128.0.0.1 。
platform.openstack.lbFloatingIP	与 API 负载均衡器关联的现有浮动 IP 地址。要使用此属性，还必须定义 platform.openstack.externalNetwork 属性。	IP 地址，如 128.0.0.1 。
platform.openstack.externalDNS	集群实例用于进行 DNS 解析的外部 DNS 服务器的 IP 地址。	一个 IP 地址列表作为字符串。例如， ["8.8.8.8", "192.168.1.12"] 。
platform.openstack.machinesSubnet	<p>集群节点使用的 RHOSP 子网的 UUID。在这个子网上创建节点和虚拟 IP (VIP) 端口。</p> <p>networking.machineNetwork 中的第一个项需要和 machinesSubnet 的值匹配。</p> <p>如果部署到自定义子网中，则无法将外部 DNS 服务器指定到 OpenShift Container Platform 安装程序。反之，把 DNS 添加到 RHOSP 的子网。</p>	作为字符串的 UUID。例如： fa806b2f-ac49-4bceb9db-124bc64209bf 。

1.3.13.6. RHOSP 部署中的自定义子网

另外，您还可以在您选择的 Red Hat OpenStack Platform (RHOSP) 子网中部署集群。子网的 GUID 作为 **install-config.yaml** 文件中的 **platform.openstack.machinesSubnet** 的值传递。

此子网被用作集群的主子网，在其上创建节点和端口。

在使用自定义子网运行 OpenShift Container Platform 安装程序前，请验证：

- 目标网络和子网可用。

- 目标子网上启用了 DHCP。
- 您可提供在目标网络上有创建端口权限的安装程序凭证。
- 如果您的网络配置需要一个路由器，它会在 RHOSP 中创建。有些配置依赖于路由器来转换浮动 IP 地址。
- 您的网络配置不依赖于供应商网络。不支持提供商网络。

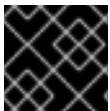


注意

默认情况下，API VIP 使用 x.x.x.5，Ingress VIP 从网络 CIDR 块获取 x.x.x.7。要覆盖这些默认值，为 DHCP 分配池以外的 **platform.openstack.apiVIP** 和 **platform.openstack.ingressVIP** 设置值。

1.3.13.7. RHOSP 的自定义 install-config.yaml 文件示例

此示例 **install-config.yaml** 展示了所有可能的 Red Hat OpenStack Platform (RHOSP) 自定义选项。



重要

此示例文件仅供参考。您必须使用安装程序来获取 **install-config.yaml** 文件。

```

apiVersion: v1
baseDomain: example.com
clusterID: os-test
controlPlane:
  name: master
  platform: {}
  replicas: 3
compute:
- name: worker
  platform:
    openstack:
      type: m1.large
  replicas: 3
metadata:
  name: example
networking:
  clusterNetwork:
  - cidr: 10.128.0.0/14
    hostPrefix: 23
  machineNetwork:
  - cidr: 10.0.0.0/16
  serviceNetwork:
  - 172.30.0.0/16
  networkType: OpenShiftSDN
platform:
  openstack:
    cloud: mycloud
    externalNetwork: external
    computeFlavor: m1.xlarge
    lbFloatingIP: 128.0.0.1

```

```
fips: false
pullSecret: '{"auths": ...}'
sshKey: ssh-ed25519 AAAA...
```

1.3.13.8. 为机器设置自定义子网

安装程序默认使用的 IP 范围可能与您在安装 OpenShift Container Platform 时创建的 Neutron 子网不匹配。如有必要，通过编辑安装配置文件来更新新机器的 CIDR 值。

先决条件

- 有 OpenShift Container Platform 安装程序生成的 **install-config.yaml** 文件。

流程

1. 在命令行中进入包含 **install-config.yaml** 的目录。
2. 在该目录中，运行脚本来编辑 **install-config.yaml** 文件或手动更新该文件：
 - 要使用脚本设置值，请运行：

```
$ python -c '
import yaml;
path = "install-config.yaml";
data = yaml.safe_load(open(path));
data["networking"]["machineNetwork"] = [{"cidr": "192.168.0.0/18"}]; ❶
open(path, "w").write(yaml.dump(data, default_flow_style=False))'
```

- ❶ 插入一个与您指定的 Neutron 子网匹配的值，如 **192.0.2.0/24**。

- 要手动设置这个值，请打开该文件并将 **networking.machineCIDR** 的值设置为与您预期的 Neutron 子网匹配的内容。

1.3.13.9. 清空计算机器池

要进行使用您自己的基础架构的安装，请将安装配置文件中的计算机器数量设置为零。之后，您可以手动创建这些机器。

先决条件

- 有 OpenShift Container Platform 安装程序生成的 **install-config.yaml** 文件。

流程

1. 在命令行中进入包含 **install-config.yaml** 的目录。
2. 在该目录中，运行脚本来编辑 **install-config.yaml** 文件或手动更新该文件：
 - 要使用脚本设置值，请运行：

```
$ python -c '
import yaml;
path = "install-config.yaml";
```



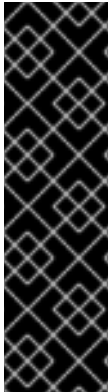
```
data = yaml.safe_load(open(path));
data["compute"][0]["replicas"] = 0;
open(path, "w").write(yaml.dump(data, default_flow_style=False))'
```

- 要手动设置值，打开文件并将 **compute.<first entry>.replicas** 的值设置为 **0**。

1.3.14. 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. 删除定义 control plane 机器的 Kubernetes 清单文件以及计算机器集：

```
$ rm -f openshift/99_openshift-cluster-api_master-machines-*.yaml openshift/99_openshift-cluster-api_worker-machineset-*.yaml
```

由于您要自行创建和管理这些资源，因此不必初始化这些资源。

- 您可以使用机器 API 来保留机器集文件来创建计算机器，但您必须更新对其的引用，以匹配您的环境。
3. 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。
4. 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
```

5. 将元数据文件的 `infraID` 键导出为环境变量：

```
$ export INFRA_ID=$(jq -r .infraID metadata.json)
```

提示

从 `metadata.json` 中提取 `infraID` 键，并将其用作您创建的所有 RHOSP 资源的前缀。通过这样做，您可以避免在同一项目中进行多个部署时的名称冲突。

1.3.15. 准备 bootstrap Ignition 文件

OpenShift Container Platform 安装过程依赖于从 bootstrap Ignition 配置文件创建的 bootstrap 机器。

编辑该文件并上传该文件。然后，创建 Red Hat OpenStack Platform (RHOSP) 用来下载主文件的辅助 bootstrap Ignition 配置文件。

先决条件

- 您有安装程序生成的 bootstrap Ignition 文件，即 `bootstrap.ign`。
- 安装程序元数据文件中的基础架构 ID 被设置为环境变量 (`$INFRA_ID`)。
 - 如果未设置变量，请参阅 [创建 Kubernetes 清单和 Ignition 配置文件](#)。
- 可以使用 HTTP(S) 来存储 bootstrap ignition 文件。
 - 所记录的步骤使用 RHOSP 镜像服务 (Glance)，但也可以使用 RHOSP Storage 服务 (Swift)、Amazon S3、内部 HTTP 服务器或临时 Nova 服务器。

流程

1. 运行以下 Python 脚本。该脚本修改 bootstrap Ignition 文件，以设置主机名，并在运行时设置 CA 证书文件：

```
import base64
import json
import os

with open('bootstrap.ign', 'r') as f:
    ignition = json.load(f)

files = ignition['storage'].get('files', [])

infra_id = os.environ.get('INFRA_ID', 'openshift').encode()
hostname_b64 = base64.standard_b64encode(infra_id + b'-bootstrap\n').decode().strip()
files.append(
{
    'path': '/etc/hostname',
    'mode': 420,
    'contents': {
        'source': 'data:text/plain;charset=utf-8;base64,' + hostname_b64
    }
})

ca_cert_path = os.environ.get('OS_CACERT', "")
if ca_cert_path:
    with open(ca_cert_path, 'r') as f:
        ca_cert = f.read().encode()
        ca_cert_b64 = base64.standard_b64encode(ca_cert).decode().strip()

    files.append(
    {
        'path': '/opt/openshift/tls/cloud-ca-cert.pem',
        'mode': 420,
        'contents': {
            'source': 'data:text/plain;charset=utf-8;base64,' + ca_cert_b64
        }
    })

ignition['storage']['files'] = files;

with open('bootstrap.ign', 'w') as f:
    json.dump(ignition, f)
```

2. 使用 RHOSP CLI，创建使用 bootstrap Ignition 文件的镜像：

```
$ openstack image create --disk-format=raw --container-format=bare --file bootstrap.ign
<image_name>
```

3. 获取镜像的详情：

```
$ openstack image show <image_name>
```

请记录 **file** 值；它需要遵循 **v2/images/<image_ID>/file** 格式。



注意

验证您创建的镜像是否活跃。

- 检索镜像服务的公共地址：

```
$ openstack catalog show image
```

- 将公共地址与镜像的 **file** 值合并，并在存储位置保存结果。位置遵循 **<image_service_public_URL>/v2/images/<image_ID>/file** 格式。

- 生成身份验证令牌并保存令牌 ID:

```
$ openstack token issue -c id -f value
```

- 将以下内容插入到名为 **\$INFRA_ID-bootstrap-ignition.json** 的文件中，并编辑位置拥有者以匹配您自己的值：

```
{
  "ignition": {
    "config": {
      "merge": [{
        "source": "<storage_url>", 1
        "httpHeaders": [{
          "name": "X-Auth-Token", 2
          "value": "<token_ID>" 3
        }]
      }]
    },
    "security": {
      "tls": {
        "certificateAuthorities": [{
          "source": "data:text/plain;charset=utf-8;base64,<base64_encoded_certificate>" 4
        }]
      }
    },
    "version": "3.1.0"
  }
}
```

- 将 **ignition.config.merge.source** 的值替换为 bootstrap Ignition 文件存储 URL。
- 在 **httpHeaders** 中将 **name** 设置为 **"X-Auth-Token"**。
- 在 **httpHeaders** 中将 **value** 设为您的令牌 ID。
- 如果 bootstrap Ignition 文件服务器使用自签名证书,请包括以 base64 编码的证书。

- 保存二级 Ignition 配置文件。

bootstrap Ignition 数据将在安装过程中传递给 RHOSP。



警告

bootstrap Ignition 文件包含敏感信息，如 **clouds.yaml** 凭证。确定您将其保存在安全的地方，并在完成安装后将其删除。

1.3.16. 在 RHOSP 上创建 control plane Ignition 配置文件

在您自己的基础架构的 Red Hat OpenStack Platform (RHOSP) 上安装 OpenShift Container Platform 需要 control plane Ignition 配置文件。您必须创建多个配置文件。



注意

与 bootstrap Ignition 配置一样，您必须明确为每个 control plane 机器定义主机名。

先决条件

- 来自安装程序元数据文件中的基础架构 ID 被设置为环境变量 (**\$INFRA_ID**)。
 - 如果未设置变量，请参阅“创建 Kubernetes 清单和 Ignition 配置文件”。

流程

- 在命令行中运行以下 Python 脚本：

```
$ for index in $(seq 0 2); do
  MASTER_HOSTNAME="$INFRA_ID-master-$index\n"
  python -c "import base64, json, sys;
  ignition = json.load(sys.stdin);
  storage = ignition.get('storage', {});
  files = storage.get('files', []);
  files.append({'path': '/etc/hostname', 'mode': 420, 'contents': {'source':
'data:text/plain;charset=utf-8;base64,' +
base64.standard_b64encode(b'$MASTER_HOSTNAME').decode().strip(), 'verification': {}},
'filesystem': 'root'});
  storage['files'] = files;
  ignition['storage'] = storage
  json.dump(ignition, sys.stdout) <master.ign >"$INFRA_ID-master-$index-ignition.json"
done
```

您现在有三个 control plane Ignition 文件：**<INFRA_ID>-master-0-ignition.json**、**<INFRA_ID>-master-1-ignition.json** 和 **<INFRA_ID>-master-2-ignition.json**。

1.3.17. 在 RHOSP 上创建网络资源

在您自己的基础架构的 Red Hat OpenStack Platform (RHOSP) 安装上创建 OpenShift Container Platform 所需的网络资源。为节省时间，可以运行提供的 Ansible playbook 来生成安全组、网络、子网、路由器和端口。

先决条件

- Python 3 已安装在您的机器上。
- 您下载了"下载 playbook 依赖项"中的模块。
- 下载了"下载安装 playbook"中的 playbook。

流程

1. 可选：为 **inventory.yaml** playbook 添加一个外部网络值：

inventory.yaml Ansible playbook 中的外部网络值示例

```
...
# The public network providing connectivity to the cluster. If not
# provided, the cluster external connectivity must be provided in another
# way.

# Required for os_api_fip, os_ingress_fip, os_bootstrap_fip.
os_external_network: 'external'
...
```



重要

如果没有为 **inventory.yaml** 文件中的 **os_external_network** 提供值，则必须确保虚拟机可以自行访问 Glance 和外部连接。

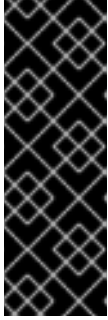
2. 可选：将外部网络和浮动 IP（FIP）地址值添加到 **inventory.yaml** playbook：

inventory.yaml Ansible playbook 中的 FIP 值示例

```
...
# OpenShift API floating IP address. If this value is non-empty, the
# corresponding floating IP will be attached to the Control Plane to
# serve the OpenShift API.
os_api_fip: '203.0.113.23'

# OpenShift Ingress floating IP address. If this value is non-empty, the
# corresponding floating IP will be attached to the worker nodes to serve
# the applications.
os_ingress_fip: '203.0.113.19'

# If this value is non-empty, the corresponding floating IP will be
# attached to the bootstrap machine. This is needed for collecting logs
# in case of install failure.
os_bootstrap_fip: '203.0.113.20'
```



重要

如果您没有为 **os_api_fip** 和 **os_ingress_fip** 定义值，则必须执行安装后的网络配置。

如果您没有为 **os_bootstrap_fip** 定义值，安装程序将无法从失败的安装中下载调试信息。

如需更多信息，请参阅"启用对环境的访问"。

3. 在命令行中，通过运行 **security-groups.yaml** playbook 来创建安全组：

```
$ ansible-playbook -i inventory.yaml security-groups.yaml
```

4. 在命令行中，通过运行 **network.yaml** playbook 来创建一个网络、子网和路由器：

```
$ ansible-playbook -i inventory.yaml network.yaml
```

5. 可选：如果要控制 Nova 服务器使用的默认解析程序，请运行 RHOSP CLI 命令：

```
$ openstack subnet set --dns-nameserver <server_1> --dns-nameserver <server_2>
"$INFRA_ID-nodes"
```

1.3.18. 在 RHOSP 上创建 bootstrap 机器

创建 bootstrap 机器，为其提供在 Red Hat OpenStack Platform (RHOSP) 上运行所需的网络访问权限。红帽提供了一个 Ansible playbook，您可运行它来简化此过程。

先决条件

- 您下载了"下载 playbook 依赖项"中的模块。
- 下载了"下载安装 playbook"中的 playbook。
- **inventory.yaml**、**common.yaml** 和 **bootstrap.yaml** Ansible playbook 位于一个通用目录中。
- 安装程序创建的 **metadata.json** 文件与 Ansible playbook 位于同一个目录中。

流程

1. 在命令行中，将工作目录改为 playbook 的位置。
2. 在命令行中运行 **bootstrap.yaml** playbook：

```
$ ansible-playbook -i inventory.yaml bootstrap.yaml
```

3. bootstrap 服务器可用后，查看日志以验证是否收到 Ignition 文件：

```
$ openstack console log show "$INFRA_ID-bootstrap"
```

1.3.19. 在 RHOSP 中创建 control plane 机器

使用您生成的 Ignition 配置文件创建三台 control plane 机器。红帽提供了一个 Ansible playbook，您可运行它来简化此过程。

先决条件

- 您下载了"下载 playbook 依赖项"中的模块。
- 下载了"下载安装 playbook"中的 playbook。
- 来自安装程序元数据文件中的基础架构 ID 被设置为环境变量 (**\$INFRA_ID**)。
- **inventory.yaml**、**common.yaml** 和 **control-plane.yaml** Ansible playbook 位于一个通用目录中。
- 您有三个在"Creating control plane Ignition 配置文件"中创建的 Ignition 文件。

流程

1. 在命令行中，将工作目录改为 playbook 的位置。
2. 如果 control plane Ignition 配置文件尚未位于工作目录中，将其复制到其中。
3. 在命令行中运行 **control-plane.yaml** playbook：

```
$ ansible-playbook -i inventory.yaml control-plane.yaml
```

4. 运行以下命令来监控 bootstrap 过程：

```
$ openshift-install wait-for bootstrap-complete
```

您会看到确认 control plane 机器正在运行并加入集群的消息：

```
INFO API v1.14.6+f9b5405 up
INFO Waiting up to 30m0s for bootstrapping to complete...
...
INFO It is now safe to remove the bootstrap resources
```

1.3.20. 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
```

1.3.21. 从 RHOSP 删除 bootstrap 资源

删除您不再需要的 bootstrap 资源。

先决条件

- 您下载了"下载 playbook 依赖项"中的模块。
- 下载了"下载安装 playbook"中的 playbook。
- **inventory.yaml**、**common.yaml** 和 **down-bootstrap.yaml** Ansible playbook 位于一个通用目录中。
- control plane 机器正在运行。
 - 如果您不知道机器的状态，请参阅"验证集群状态"。

流程

1. 在命令行中，将工作目录改为 playbook 的位置。
2. 在命令行中运行 **down-bootstrap.yaml** playbook：

```
$ ansible-playbook -i inventory.yaml down-bootstrap.yaml
```

bootstrap 端口、服务器和浮动 IP 地址会被删除。



警告

如果您之前没有禁用 bootstrap Ignition 文件 URL，现在需要禁用。

1.3.22. 在 RHOSP 上创建计算机

启动 control plane 后，创建计算机。红帽提供了一个 Ansible playbook，您可运行它来简化此过程。

先决条件

- 您下载了"下载 playbook 依赖项"中的模块。

- 下载了"下载安装 playbook"中的 playbook。
- **inventory.yaml**、**common.yaml** 和 **compute-nodes.yaml** Ansible playbook 位于一个通用目录中。
- 安装程序创建的 **metadata.json** 文件与 Ansible playbook 位于同一个目录中。
- control plane 处于活跃状态。

流程

1. 在命令行中，将工作目录改为 playbook 的位置。
2. 在命令行中运行 playbook:

```
$ ansible-playbook -i inventory.yaml compute-nodes.yaml
```

后续步骤

- 批准机器的证书签名请求。

1.3.23. 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。您可能在列表中看到更多已批准的 CSR。

- 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-bootstrap** 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。

- 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"}\n{{end}}{{end}}' | xargs oc adm certificate approve
```

6. 批准所有客户端和服务器的 CSR 后，节点将处于 **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节点转换为 **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](#)。

1.3.24. 验证安装是否成功

验证 OpenShift Container Platform 安装已完成。

先决条件

- 有安装程序 (`openshift-install`)

流程

- 在命令行中运行：

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

程序输出控制台 URL 以及管理员的登录信息。

1.3.25. 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](#)。

1.3.26. 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](#)。
- 如果您需要启用对节点端口的外部访问，[请使用节点端口配置集群流量](#)。
- 如果您没有将 RHOSP 配置为使用浮动 IP 地址接受应用程序流量，[使用浮动 IP 地址配置 RHOSP 访问](#)。

1.4. 在您自己的基础架构上带有 KURYR 的 OPENSTACK 上安装集群

在 OpenShift Container Platform 版本 4.6 中，您可以在运行于用户自备的基础架构上的 Red Hat OpenStack Platform (RHOSP) 上安装集群。

通过利用您自己的基础架构，您可以将集群与现有的基础架构进行集成。和安装程序自备的安装方式相比，这个过程需要用户进行更多操作，因为您必须创建所有 RHOSP 资源，如 Nova 服务器、Neutron 端口和安全组。红帽提供了 Ansible playbook 来帮助您完成部署过程。

1.4.1. 先决条件

- 查看有关 [OpenShift Container Platform 安装和更新](#) 流程的详细信息。
 - 在 *Available platforms* 部分验证 OpenShift Container Platform 4.6 是否与您的 RHOSP 版本兼容。您还可以查看 [OpenShift Container Platform 在 RHOSP 中的支持](#) 来比较不同版本的平台支持。
- 验证您的网络配置不依赖于供应商网络。不支持提供商网络。
- 具有要安装 OpenShift Container Platform 的 RHOSP 帐户
- 在您运行安装程序的机器中，有：

- 用来保存在安装过程中创建的文件的一个单一目录
- Python 3

1.4.2. 关于 Kuryr SDN

Kuryr 是一个容器网络接口 (CNI) 插件解决方案，它使用 **Neutron** 和 **Octavia** Red Hat OpenStack Platform (RHOSP) 服务来为 pod 和服务提供网络。

Kuryr 和 OpenShift Container Platform 的集成主要针对在 RHOSP VM 上运行的 OpenShift Container Platform 集群设计。Kuryr 通过将 OpenShift Container Platform pod 插入到 RHOSP SDN 来提高网络性能。另外，它还提供 pod 和 RHOSP 虚拟实例间的互联性。

Kuryr 组件作为 pod 在 OpenShift Container Platform 中安装，使用 **openshift-kuryr** 命名空间：

- **kuryr-controller** - 在一个 **master** 节点上安装的单个服务实例。这在 OpenShift Container Platform 中建模为一个 **Deployment** 对象。
- **kuryr-cni** - 在每个 OpenShift Container Platform 节点上安装并配置 Kuryr 作为 CNI 驱动的容器。这在 OpenShift Container Platform 中建模为一个 **DaemonSet** 对象。

Kuryr 控制器监控 OpenShift Container Platform API 服务器中的 pod、服务和命名空间创建、更新和删除事件。它将 OpenShift Container Platform API 调用映射到 Neutron 和 Octavia 中的对应对象。这意味着，实现了 Neutron 中继端口功能的每个网络解决方案都可以通过 Kuryr 支持 OpenShift Container Platform。这包括开源解决方案，比如 Open vSwitch (OVS) 和 Open Virtual Network (OVN)，以及 Neutron 兼容的商业 SDN。

建议在封装的 RHOSP 租户网络上部署 OpenShift Container Platform 时使用 Kuryr，以避免出现重复封装，例如通过 RHOSP 网络运行封装的 OpenShift Container Platform SDN。

如果您使用供应商网络或租户 VLAN，则不需要使用 Kuryr 来避免重复封装。虽然性能上的优势微不足道，但根据您的配置，使用 Kuryr 避免两个覆盖可能仍然有用。

在完足以下所有条件的部署中不建议使用 Kuryr：

- RHOSP 版本早于 16
- 部署使用 UDP 服务，或者在几个 hypervisor 上使用大量 TCP 服务。

或

- **ovn-octavia** Octavia 驱动被禁用。
- 部署在几个 hypervisor 中使用了大量的 TCP 服务。

1.4.3. 在带有 Kuryr 的 OpenStack 上安装 OpenShift Container Platform 的资源指南

当使用 Kuryr SDN 时，pod、服务、命名空间和网络策略会使用来自 RHOSP 配额的资源，这会增加最低要求。除了默认安装需要满足的要求，Kuryr 还有一些额外的要求。

使用以下配额来满足集群的默认最低要求：

表 1.19. 带有 Kuryr 的 RHOSP 上默认 OpenShift Container Platform 集群的建议资源

资源	值
浮动 IP 地址	3 - 加上预期的 LoadBalancer 类型服务的数量
端口	1500 - 每个 Pod 需要 1 个
路由器	1
子网	250 - 每个命名空间/项目需要 1 个
网络	250 - 每个命名空间/项目需要 1 个
RAM	112 GB
vCPUs	28
卷存储	275 GB
实例	7
安全组	250 - 每个服务和每个 NetworkPolicy 需要 1 个
安全组规则	1000
负载均衡器	100 - 每个服务需要 1 个
负载均衡器侦听程序	500 - 每个服务公开端口需要 1 个
负载均衡器池	500 - 每个服务公开端口需要 1 个

集群或许能使用少于推荐数量的资源来运作，但其性能无法保证。



重要

如果 RHOSP 对象存储 (Swift) 可用，并由具有 **swiftoperator** 角色的用户帐户执行，它会作为 OpenShift Container Platform 镜像 registry 的默认后端。在这种情况下，卷存储需要有 175GB。根据镜像 registry 的大小，Swift 空间要求会有所不同。



重要

如果您使用带有 Amphora 驱动而不是 OVN Octavia 驱动的 Red Hat OpenStack Platform (RHOSP) 版本 16，则安全组会与服务帐户而不是用户项目关联。

在设置资源时请考虑以下几点：

- 需要的端口数量会大于 pod 的数量。Kuryr 使用端口池来预创建端口以供 pod 使用，用于加快 pod 的启动时间。

- 每个网络策略都映射到 RHOSP 安全组中，并根据 **NetworkPolicy** 规格将一个或多个规则添加到安全组中。
- 每个服务都映射到一个 RHOSP 负载均衡器中。在估算配额所需安全组数时，请考虑此要求。如果您使用 RHOSP 版本 15 或更早版本，或者使用 **ovn-octavia** 驱动，则每个负载均衡器都有一个带有用户项目的安全组。
- 配额不考虑负载均衡器资源（如 VM 资源），但您必须在决定 RHOSP 部署的大小时考虑这些资源。默认安装将有超过 50 个负载均衡器，集群必须可以容纳它们。如果您使用启用 OVN Octavia 驱动程序的 RHOSP 版本 16，则只生成一个负载均衡器虚拟机；服务通过 OVN 流平衡负载。

OpenShift Container Platform 部署由 control plane 机器、计算机器和 bootstrap 机器组成。

要启用 Kuryr SDN，您的环境必须满足以下要求：

- 运行 RHOSP 13+。
- 具有 Octavia 的 Overcloud。
- 使用 Neutron Trunk 端口扩展。
- 如果使用 ML2/OVS Neutron 驱动而不是 **ovs-hybrid**，则请使用 **openvswitch** 防火墙驱动。

1.4.3.1. 增加配额

使用 Kuryr SDN 时，您必须提高配额以满足 pod、Services、namespaces 和网络策略所使用的 Red Hat OpenStack Platform (RHOSP) 资源要求。

流程

- 运行以下命令为项目增加配额：

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

1.4.3.2. 配置 Neutron

Kuryr CNI 利用 Neutron Trunks 扩展来将容器插入 Red Hat OpenStack Platform (RHOSP) SDN，因此您必须使用 **trunks** 扩展才可以使 Kuryr 正常工作。

另外，如果您使用默认的 ML2/OVS Neutron 驱动程序，防火墙必须设为 **openvswitch** 而不是 **ovs_hybrid**，以便在中继子端口上强制实施安全组，同时 Kuryr 可以正确处理网络策略。

1.4.3.3. 配置 Octavia

Kuryr SDN 使用 Red Hat OpenStack Platform (RHOSP) 的 Octavia LBaaS 来实现 OpenShift Container Platform 服务。因此，您必须在 RHOSP 上安装和配置 Octavia 组件以使用 Kuryr SDN。

要启用 Octavia，您必须在安装 RHOSP Overcloud 的过程中包括 Octavia 服务，如果 Overcloud 已存在则需要升级 Octavia 服务。以下启用 Octavia 的步骤适用于新的 Overcloud 安装或 Overcloud 更新。



注意

以下步骤只包括在部署 RHOSP 时需要处理 Octavia 部分的信息。请注意 `registry` 可能会不同。

这个示例使用本地的 registry。

流程

1. 如果您使用本地 registry，请创建一个模板来将镜像上传到 registry。例如：

```
(undercloud) $ openstack overcloud container image prepare \
-e /usr/share/openstack-tripleo-heat-templates/environments/services-docker/octavia.yaml \
--namespace=registry.access.redhat.com/rhosp13 \
--push-destination=<local-ip-from-undercloud.conf>:8787 \
--prefix=openstack- \
--tag-from-label {version}-{release} \
--output-env-file=/home/stack/templates/overcloud_images.yaml \
--output-images-file /home/stack/local_registry_images.yaml
```

2. 验证 `local_registry_images.yaml` 文件是否包含 Octavia 镜像。例如：

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



注意

Octavia 容器版本根据所安装的特定 RHOSP 版本的不同而有所不同。

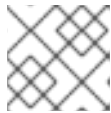
3. 将 `registry.redhat.io` 中的容器镜像拉取到 Undercloud 节点：

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

这可能需要一些时间，具体要看您的网络速度和 Undercloud 使用的磁盘。

4. 由于 Octavia 负载均衡器是用来访问 OpenShift Container Platform API，所以您必须增加它们的监听程序的默认超时时间。默认超时为 50 秒。通过将以下文件传递给 Overcloud deploy 命令，将超时时间增加到 20 分钟：

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

**注意**

RHOSP 13.0.13+ 不需要这一步。

5. 使用 Octavia 安装或更新 overcloud 环境：

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

**注意**

这个命令只包含与 Octavia 相关的文件，它根据您具体的 RHOSP 安装而有所不同。如需更多信息，请参阅 RHOSP 文档。有关自定义 Octavia 安装的详情请参考 [使用 Director 安装 Octavia](#)。

**注意**

当利用 Kuryr SDN 时，Overcloud 安装需要 Neutron **trunk** 扩展。这在 director 部署中默认可用。当 Neutron 后端是 ML2/OVS 时，使用 **openvswitch** 防火墙而不是默认的 **ovs-hybrid**。如果后端为 ML2/OVN，则不需要修改。

6. 在早于 13.0.13 的 RHOSP 版本中，在创建项目后将项目 ID 添加到 **octavia.conf** 配置文件中。

- 要跨服务实施网络策略，比如网络流量会通过 Octavia 负载均衡器时，您必须确保 Octavia 在用户项目中创建 Amphora VM 安全组。这可确保所需的 LoadBalancer 安全组属于该项目，并可将其更新为强制实施服务隔离。

**注意**

在 RHOSP 13.0.13 或更高版本中不需要此操作。

Octavia 实施新的 ACL API，限制对负载均衡器 VIP 的访问。

a. 获取项目 ID

```
$ openstack project show <project>
```

输出示例

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

b. 将项目 ID 添加到控制器的 **octavia.conf** 中。

i. Source **stackrc** 文件：

```
$ source stackrc # Undercloud credentials
```

ii. 列出 Overcloud 控制器。

```
$ openstack server list
```

输出示例

```
+-----+-----+-----+-----+
| ID              | Name      | Status | Networks |
| Image          | Flavor   |        |          |
+-----+-----+-----+-----+
| 6bef8e73-2ba5-4860-a0b1-3937f8ca7e01 | controller-0 | ACTIVE | ctlplane=192.168.24.8 |
| overcloud-full | controller |
| dda3173a-ab26-47f8-a2dc-8473b4a67ab9 | compute-0   | ACTIVE |  |
| overcloud-full | compute   |
+-----+-----+-----+-----+
```

iii. SSH 到控制器。

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

iv. 编辑 **octavia.conf** 文件，将项目添加到 Amphora 安全组存在于用户账户的项目列表中。

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

c. 重启 Octavia worker 以便重新加载配置。

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



注意

根据您的 RHOSP 环境，Octavia 可能不支持 UDP 侦听程序。如果您在 RHOSP 版本 13.0.13 或更早版本使用 Kuryr SDN，则不支持 UDP 服务。RHOSP 版本 16 或更高版本支持 UDP。

1.4.3.3.1. Octavia OVN 驱动程序

Octavia 通过 Octavia API 支持多个供应商驱动程序。

要查看所有可用的 Octavia 提供程序驱动，请在命令行中输入：

```
$ openstack loadbalancer provider list
```

输出示例

```
+-----+-----+
| name | description |
+-----+-----+
| amphora | The Octavia Amphora driver. |
| octavia | Deprecated alias of the Octavia Amphora driver. |
| ovn | Octavia OVN driver. |
+-----+-----+
```

从 RHOSP 版本 16 开始，Octavia OVN 供应商驱动程序 (**ovn**) 在 RHOSP 部署的 OpenShift Container Platform 上被支持。

ovn 是 Octavia 和 OVN 提供的负载均衡集成驱动。它支持基本负载均衡功能，并基于 OpenFlow 规则。在使用 OVN Neutron ML2 的部署中，Director 会在 Octavia 中自动启用该驱动程序。

Amphora 供应商驱动程序是默认驱动程序。如果启用了 **ovn**，Kuryr 将使用它。

如果 Kuryr 使用 **ovn** 而不是 Amphora，则可提供以下优点：

- 资源要求更低 Kuryr 不需要为每个服务都提供一个负载均衡器虚拟机。
- 网络延迟会降低。
- 通过对每个服务使用 OpenFlow 规则而不是 VM 来提高服务创建速度。
- 跨所有节点的分布式负载均衡操作，而不是集中到 Amphora 虚拟机中。

1.4.3.4. 已知使用 Kuryr 安装的限制

将 OpenShift Container Platform 与 Kuryr SDN 搭配使用有一些已知的限制。

RHOSP 常规限制

带有 Kuryr SDN 的 OpenShift Container Platform 不支持带有类型 **NodePort** 的 **Service** 对象。

如果机器子网没有连接到路由器，或者子网已连接，但路由器没有设置外部网关，Kuryr 无法为类型为 **LoadBalancer** 的 **Service** 对象创建浮动 IP。

- 在 **Service** 对象上配置 **sessionAffinity=ClientIP** 属性无效。Kuryr 不支持此设置。

RHOSP 版本限制

使用带有 Kuryr SDN 的 OpenShift Container Platform 有一些限制，具体取决于 RHOSP 版本。

- RHOSP 16 之前的版本使用默认 Octavia 负载均衡器驱动程序 (Amphora)。此驱动要求在每个 OpenShift Container Platform 服务中部署一个 Amphora 负载均衡器虚拟机。创建太多的服务会导致您耗尽资源。
如果以后版本的 RHOSP 部署中禁用了 OVN Octavia 驱动程序，则也会使用 Amphora 驱动。它们对资源的要求和早期版本 RHOSP 相同。

- Octavia RHOSP 13.0.13 之前的版本不支持 UDP 侦听程序。因此，OpenShift Container Platform UDP 服务不被支持。
- Octavia RHOSP 13.0.13 之前的版本无法侦听同一端口上的多个协议。不支持将同一端口暴露给不同协议的服务，比如 TCP 和 UDP。
- Kuryr SDN 不支持由服务自动取消闲置。

RHOSP 环境限制

使用取决于您的部署环境的 Kuryr SDN 会有一些限制。

由于 Octavia 缺少对 UDP 协议和多个监听器的支持，如果 rhosp 版本早于 13.0.13，Kuryr 会强制 pod 在 DNS 解析中使用 TCP，如果：

在 Go 版本 1.12 及更早的版本中，通过 CGO 支持被禁用的模式编译的应用程序只使用 UDP。在这种情况下，native Go 解析器无法识别 **resolv.conf** 中的 **use-vc** 选项，它控制 DNS 解析是否强制使用 TCP。因此，UDP 仍会被用来解析 DNS，这将导致失败。

要确保 TCP 强制使用是允许的，在编译应用程序使把环境变量 **CGO_ENABLED** 设定为 **1**（如 **CGO_ENABLED=1**），或者不使用这个变量。

在 Go 版本 1.13 及之后的版本中，如果使用 UDP 的 DNS 解析失败，则会自动使用 TCP。



注意

基于 musl 的容器，包括基于 Alpine 的容器，不支持 **use-vc** 选项。

RHOSP 升级限制

作为 RHOSP 升级过程的结果，可能会更改 Octavia API，并可能需要升级到用于负载均衡器的 Amphora 镜像。

您可以单独处理 API 更改。

如果升级了 Amphora 镜像，RHOSP Operator 可使用两种方式处理现有的负载均衡器虚拟机：

- 通过触发[负载均衡器故障切换](#)来升级每个虚拟机。
- 将升级虚拟机的职责留给用户。

如果运算符使用第一个选项，在故障切换过程中可能会有短暂的停机时间。

如果 Operator 采用第二个选项，现有负载均衡器将不支持升级的 Octavia API 功能，比如 UDP 侦听程序。在这种情况下，用户必须重新创建自己的服务以使用这些功能。



重要

如果 OpenShift Container Platform 检测到支持 UDP 负载均衡的新 Octavia 版本，它会自动重新创建 DNS 服务。服务重新创建可确保服务默认支持 UDP 负载均衡。

这个重新创建会导致 DNS 服务大约停机一分钟。

1.4.3.5. control plane 机器

默认情况下，OpenShift Container Platform 安装过程会创建三台 control plane 机器。

每台机器都需要：

- 来自 RHOSP 配额的实例
- 来自 RHOSP 配额的端口
- 至少有 16 GB 内存、4 个 vCPU 和 100 GB 存储空间的类别

1.4.3.6. 计算机器

默认情况下，OpenShift Container Platform 安装过程会创建三台计算机器。

每台机器都需要：

- 来自 RHOSP 配额的实例
- 来自 RHOSP 配额的端口
- 至少有 8 GB 内存、2 个 vCPU 和 100 GB 存储空间的类别

提示

计算机器托管您在 OpenShift Container Platform 上运行的应用程序；运行数量应尽可能多。

1.4.3.7. bootstrap 机器

在安装时，会临时置备 bootstrap 机器来支持 control plane。生产控制平面就绪后，bootstrap 机器会被取消置备。

bootstrap 机器需要：

- 来自 RHOSP 配额的实例
- 来自 RHOSP 配额的端口
- 至少有 16 GB 内存、4 个 vCPU 和 100 GB 存储空间的类别

1.4.4. 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。

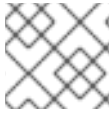


重要

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。

1.4.5. 下载 **playbook** 的依赖项

简化用户置备基础架构安装过程的 Ansible **playbook** 需要几个 Python 模块。在您要运行安装程序的机器上添加模块的仓库，然后下载它们。



注意

这些说明假设您使用 Red Hat Enterprise Linux (RHEL) 8。

先决条件

- Python 3 已安装在您的机器上。

流程

1. 在命令行中添加软件仓库：

- a. 使用 Red Hat Subscription Manager 注册：

```
$ sudo subscription-manager register # If not done already
```

- b. 获取最新的订阅数据：

```
$ sudo subscription-manager attach --pool=$YOUR_POOLID # If not done already
```

- c. 禁用当前的软件仓库：

```
$ sudo subscription-manager repos --disable=* # If not done already
```

- d. 添加所需的软件仓库：

```
$ sudo subscription-manager repos \
--enable=rhel-8-for-x86_64-baseos-rpms \
--enable=openstack-16-tools-for-rhel-8-x86_64-rpms \
--enable=ansible-2.9-for-rhel-8-x86_64-rpms \
--enable=rhel-8-for-x86_64-appstream-rpms
```

2. 安装模块：

```
$ sudo yum install python3-openstackclient ansible python3-openstacksdk python3-netaddr
```

3. 确保 **python** 命令指向 **python3**:

```
$ sudo alternatives --set python /usr/bin/python3
```

1.4.6. 下载安装 **playbook**

下载 Ansible **playbook**，可用于在您自己的 Red Hat OpenStack Platform (RHOSP) 基础架构上安装 OpenShift Container Platform。

先决条件

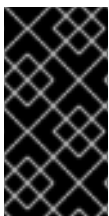
- curl 命令行工具可在您的机器上找到。

流程

- 要将 playbook 下载到您的工作目录中，请从命令行运行以下脚本：

```
$ xargs -n 1 curl -O <<< '
  https://raw.githubusercontent.com/openshift/installer/release-
4.6/upi/openstack/bootstrap.yaml
  https://raw.githubusercontent.com/openshift/installer/release-
4.6/upi/openstack/common.yaml
  https://raw.githubusercontent.com/openshift/installer/release-
4.6/upi/openstack/compute-nodes.yaml
  https://raw.githubusercontent.com/openshift/installer/release-4.6/upi/openstack/control-
plane.yaml
  https://raw.githubusercontent.com/openshift/installer/release-
4.6/upi/openstack/inventory.yaml
  https://raw.githubusercontent.com/openshift/installer/release-
4.6/upi/openstack/network.yaml
  https://raw.githubusercontent.com/openshift/installer/release-4.6/upi/openstack/security-
groups.yaml
  https://raw.githubusercontent.com/openshift/installer/release-4.6/upi/openstack/down-
bootstrap.yaml
  https://raw.githubusercontent.com/openshift/installer/release-4.6/upi/openstack/down-
compute-nodes.yaml
  https://raw.githubusercontent.com/openshift/installer/release-4.6/upi/openstack/down-
control-plane.yaml
  https://raw.githubusercontent.com/openshift/installer/release-4.6/upi/openstack/down-
load-balancers.yaml
  https://raw.githubusercontent.com/openshift/installer/release-4.6/upi/openstack/down-
network.yaml
  https://raw.githubusercontent.com/openshift/installer/release-4.6/upi/openstack/down-
security-groups.yaml
  https://raw.githubusercontent.com/openshift/installer/release-4.6/upi/openstack/down-
containers.yaml'
```

playbook 下载到您的机器中。



重要

在安装过程中，您可以修改 playbook 来配置部署。

在集群生命周期中保留所有 playbook。您必须具有 playbook，才能从 RHOSP 中删除 OpenShift Container Platform 集群。



重要

您在 **bootstrap.yaml**、**compute-nodes.yaml**、**control-plane.yaml**、**network.yaml** 和 **security-groups.yaml** 文件中进行的任何改变都需要与带有 **down-** 前缀的对应的 playbook 相匹配。例如，对 **bootstrap.yaml** 文件的编辑也必须反映在 **down-bootstrap.yaml** 文件中。如果没有编辑这两个文件，则支持的删除集群过程将失败。

1.4.7. 获取安装程序

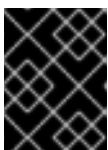
在安装 OpenShift Container Platform 之前，将安装文件下载到本地计算机上。

先决条件

- 运行 Linux 或 macOS 的计算机，本地磁盘空间为 500 MB

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。

1.4.8. 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。

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。

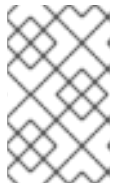
流程

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。

1.4.9. 创建 Red Hat Enterprise Linux CoreOS (RHCOS) 镜像

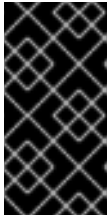
OpenShift Container Platform 安装程序要求 Red Hat OpenStack Platform (RHOSP) 集群中有 Red Hat Enterprise Linux CoreOS (RHCOS) 镜像。检索最新的 RHCOS 镜像，然后使用 RHOSP CLI 上传该镜像。

先决条件

- 已安装了 RHOSP CLI。

流程

1. 登录到红帽客户门户网站的[产品下载页](#)。
2. 在 **Version** 下，为 Red Hat Enterprise Linux (RHEL) 8 选择 OpenShift Container Platform 4.6 的最新发行版本。



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。如果可用，请使用与 OpenShift Container Platform 版本匹配的镜像版本。

3. 下载 *Red Hat Enterprise Linux CoreOS (RHCOS) - OpenStack Image (QCOW)* 。
4. 解压镜像。



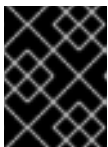
注意

您必须解压 RHOSP 镜像，然后集群才能使用它。下载的文件名可能不包含压缩扩展名，如 **.gz** 或 **.tgz**。要找出是否以及如何压缩文件，请在命令行中输入：

```
$ file <name_of_downloaded_file>
```

5. 从您下载的镜像，使用 RHOSP CLI 在集群中创建名为 **rhcos** 的镜像：

```
$ openstack image create --container-format=bare --disk-format=qcow2 --file rhcos-
${RHCOS_VERSION}-openstack.qcow2 rhcos
```



重要

根据您的 RHOSP 环境，可能需要使用 **.raw** 或 **.qcow2** 格式下载镜像。如果使用 Ceph，则必须使用 **.raw** 格式。



警告

如果安装程序发现多个同名的镜像，它会随机选择其中之一。为避免这种行为，请在 RHOSP 中为资源创建唯一名称。

将镜像上传到 RHOSP 后，就可以被安装程序使用。

1.4.10. 验证外部网络访问

OpenShift Container Platform 安装进程需要外部网络访问权限。您必须为其提供外部网络值，否则部署会失败。在运行安装进程前，请验证 Red Hat OpenStack Platform (RHOSP) 中是否存在具有外部路由器类型的网络。

先决条件

- 将 OpenStack 联网服务配置为使用 DHCP 代理转发实例 DNS 查询

流程

1. 使用 RHOSP CLI 验证“外部”网络的名称和 ID：

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

输出示例

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

网络列表中会显示具有外部路由器类型的网络。如果最少有一个没有，请参阅 [创建默认浮动 IP 网络](#)和 [创建默认供应商网络](#)。



注意

如果启用了 Neutron 中继服务插件，则默认创建中继端口。如需更多信息，请参阅 [Neutron 中继端口](#)。

1.4.11. 启用对环境的访问

在部署时，所有 OpenShift Container Platform 机器都是在 Red Hat OpenStack Platform (RHOSP) 租户网络中创建的。因此，大多数 RHOSP 部署中都无法直接访问它们。

您可以在安装过程中使用浮动 IP 地址 (FIP) 来配置 OpenShift Container Platform API 和应用程序访问。您也可以在没有配置 FIP 的情况下完成安装，但安装程序不会配置一种从外部访问 API 或应用程序的方法。

1.4.11.1. 启用通过浮动 IP 地址进行访问

创建浮动 IP(FIP)地址，用于从外部访问 OpenShift Container Platform API、集群应用程序和 bootstrap 过程。

流程

1. 使用 Red Hat OpenStack Platform (RHOSP) CLI，创建 API FIP：

```
$ openstack floating ip create --description "API <cluster_name>.<base_domain>"
<external_network>
```

2. 使用 Red Hat OpenStack Platform (RHOSP) CLI，创建应用程序或 Ingress，FIP：

```
$ openstack floating ip create --description "Ingress <cluster_name>.<base_domain>"
<external_network>
```

3. 使用 Red Hat OpenStack Platform (RHOSP) CLI 创建 bootstrap FIP:

```
$ openstack floating ip create --description "bootstrap machine" <external_network>
```

4. 向用于 API 和 Ingress FIP 的 DNS 服务器添加符合这些模式的记录：

```
api.<cluster_name>.<base_domain>. IN A <API_FIP>
*.apps.<cluster_name>.<base_domain>. IN A <apps_FIP>
```

注意

如果您不控制 DNS 服务器，您可以通过将集群域名（如以下内容）添加到 `/etc/hosts` 文件中来访问集群：

- `<api_floating_ip> api.<cluster_name>.<base_domain>`
- `<application_floating_ip> grafana-openshift-monitoring.apps.<cluster_name>.<base_domain>`
- `<application_floating_ip> prometheus-k8s-openshift-monitoring.apps.<cluster_name>.<base_domain>`
- `<application_floating_ip> oauth-openshift.apps.<cluster_name>.<base_domain>`
- `<application_floating_ip> console-openshift-console.apps.<cluster_name>.<base_domain>`
- `application_floating_ip integrate-oauth-server-openshift-authentication.apps.<cluster_name>.<base_domain>`

`/etc/hosts` 文件中的集群域名授予对本地集群的 Web 控制台和监控界面的访问权限。您还可以使用 `kubectl` 或 `oc`。您可以使用指向 `<application_floating_ip>` 的额外条目来访问用户应用程序。此操作使 API 和应用程序可供您访问，不适用于生产部署，但允许对开发和测试进行安装。

5. 将 FIP 添加到 `inventory.yaml` 文件，作为以下变量的值：

- `os_api_fip`
- `os_bootstrap_fip`
- `os_ingress_fip`

如果使用这些值，还必须在 `inventory.yaml` 文件中输入一个外部网络作为 `os_external_network` 变量的值。

提示

您可以通过分配浮动 IP 地址并更新防火墙配置，使 OpenShift Container Platform 资源在集群之外可用。

1.4.11.2. 完成没有浮动 IP 地址的安装

您可以在不提供浮动 IP 地址的情况下在 Red Hat OpenStack Platform (RHOSP) 上安装 OpenShift Container Platform。

在 `inventory.yaml` 文件中，不要定义以下变量：

- `os_api_fip`
- `os_bootstrap_fip`
- `os_ingress_fip`

如果无法提供外部网络，也可以将 `os_external_network` 留空。如果没有为 `os_external_network` 提供值，则不会为您创建路由器。如果没有额外的操作，安装程序将无法从 Glance 检索镜像。之后在安装过程中，当您创建网络资源时，必须自行配置外部连接。

如果您使用 `wait-for` 命令从因为缺少浮动 IP 地址或名称解析而无法访问集群 API 的系统中运行安装程序时，安装会失败。要防止安装失败，可以使用代理网络或者从与您的机器位于同一网络的系统中运行安装程序。



注意

您可以通过为 API 和 Ingress 端口创建 DNS 记录来启用名称解析。例如：

```
api.<cluster_name>.<base_domain>. IN A <api_port_IP>
*.apps.<cluster_name>.<base_domain>. IN A <ingress_port_IP>
```

如果您不控制 DNS 服务器，可以改为将记录添加到 `/etc/hosts` 文件中。此操作使 API 可供您自己访问，不适合于生产部署。这可用于进行开发和测试的安装。

1.4.12. 为安装程序定义参数

OpenShift Container Platform 安装程序依赖于一个名为 `clouds.yaml` 的文件。该文件描述了 Red Hat OpenStack Platform (RHOSP) 配置参数，包括项目名称、登录信息和授权服务 URL。

流程

1. 创建 `clouds.yaml` 文件：

- 如果您的 RHOSP 发行版包含 Horizon web UI，请在该 UI 中生成 `clouds.yaml` 文件。



重要

请记住在 `auth` 字段中添加密码。您也可以把 secret 保存在 `clouds.yaml` 以外的一个独立的文件中。

- 如果您的 RHOSP 发行版不包含 Horizon Web UI，或者您不想使用 Horizon，请自行创建该文件。如需有关 `clouds.yaml` 的详细信息，请参阅 RHOSP 文档中的 [配置文件](#)。

```
clouds:
  shiftstack:
    auth:
      auth_url: http://10.10.14.42:5000/v3
      project_name: shiftstack
      username: shiftstack_user
      password: XXX
      user_domain_name: Default
      project_domain_name: Default
```

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

2. 如果您的 RHOSP 安装使用自签名证书颁发机构 (CA) 证书进行端点身份验证：

- a. 将 CA 文件复制到您的机器中。
- b. 将机器添加到证书颁发机构信任捆绑包中：

```
$ sudo cp ca.crt.pem /etc/pki/ca-trust/source/anchors/
```

- c. 更新信任捆绑包：

```
$ sudo update-ca-trust extract
```

- d. 将 **cacerts** 键添加到 **clouds.yaml** 文件。该值必须是到 CA 证书的绝对路径，则其可以被非根用户访问：

```
clouds:
  shiftstack:
    ...
  cacert: "/etc/pki/ca-trust/source/anchors/ca.crt.pem"
```

提示

使用自定义 CA 证书运行安装程序后，您可以通过编辑 **cloud-provider-config** keymap 中的 **ca-cert.pem** 键的值来更新证书。在命令行中运行：

```
$ oc edit configmap -n openshift-config cloud-provider-config
```

3. 将 **clouds.yaml** 文件放在以下位置之一：

- a. **OS_CLIENT_CONFIG_FILE** 环境变量的值
- b. 当前目录
- c. 特定于 Unix 的用户配置目录，如 **~/.config/openstack/clouds.yaml**
- d. 特定于 Unix 的站点配置目录，如 **/etc/openstack/clouds.yaml**
安装程序会按照以上顺序搜索 **clouds.yaml**。

1.4.13. 创建安装配置文件

您可以自定义在 Red Hat OpenStack Platform (RHOSP) 上安装的 OpenShift Container Platform 集群。

先决条件

- 获取 OpenShift Container Platform 安装程序以及集群的 pull secret。

流程

1. 创建 `install-config.yaml` 文件。
 - a. 更改到包含安装程序的目录，再运行以下命令：

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

- 1** 对于 `<installation_directory>`，请指定用于保存安装程序所创建的文件目录名称。



重要

指定一个空目录。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。

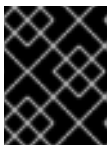
- b. 在提示符处，提供您的云的配置详情：
 - i. 可选：选择用来访问集群机器的 SSH 密钥。



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。

- ii. 选择 `openstack` 作为目标平台。
 - iii. 指定用于安装集群的 Red Hat OpenStack Platform (RHOSP) 外部网络名称。
 - iv. 指定用于从外部访问 OpenShift API 的浮动 IP 地址。
 - v. 指定至少有 16 GB RAM 用于 control plane 和计算节点的 RHOSP 类别。
 - vi. 选择集群要部署到的基域。所有 DNS 记录都将是这个基域的子域，并包含集群名称。
 - vii. 为集群输入一个名称。名称不能多于 14 个字符。
 - viii. 粘贴 [Red Hat OpenShift Cluster Manager 中的 pull secret](#)。
2. 修改 `install-config.yaml` 文件。您可以在 [安装配置参数](#) 部分中找到有关可用参数的更多信息。
 3. 备份 `install-config.yaml` 文件，以便用于安装多个集群。



重要

`install-config.yaml` 文件会在安装过程中消耗掉。如果要重复使用此文件，必须现在备份。

现在，文件 `install-config.yaml` 位于您指定的目录中。

1.4.14. 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。

1.4.14.1. 所需的配置参数

下表描述了所需的安装配置参数：

表 1.20. 所需的参数

参数	描述	值
<code>apiVersion</code>	<code>install-config.yaml</code> 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
<code>baseDomain</code>	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 <code>baseDomain</code> 和 <code>metadata.name</code> 参数值的组合，其格式为 <code><metadata.name>.<baseDomain></code> 。	完全限定域名或子域名，如 example.com 。
<code>metadata</code>	Kubernetes 资源 ObjectMeta ，其中只消耗 <code>name</code> 参数。	对象
<code>metadata.name</code>	集群的名称。集群的 DNS 记录是 <code>{{.metadata.name}}</code> 。 <code>{{.baseDomain}}</code> 的子域。	小写字母、连字符(-)和句点(.)的字符串，如 dev 。该字符串长度必须为 14 个字符或更少。
<code>platform</code>	执行安装的具体平台配置： aws 、 baremetal 、 azure 、 openstack 、 ovirt 、 vsphere 。有关 <code>platform.<platform></code> 参数的额外信息，请参考下表来了解您的具体平台。	对象

参数	描述	值
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>

1.4.14.2. 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 1.21. 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： <pre>networking: clusterNetwork: - cidr: 10.128.0.0/14 hostPrefix: 23</pre>
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。

参数	描述	值
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： <pre>networking: serviceNetwork: - 172.30.0.0/16</pre>
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。	一个对象数组。例如： <pre>networking: machineNetwork: - cidr: 10.0.0.0/16</pre>
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 。  <p>注意</p> <p>将 networking.machineNetwork 设置为与首选 NIC 所在的 CIDR 匹配。</p>

1.4.14.3. 可选配置参数

下表描述了可选安装配置参数：

表 1.22. 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。

参数	描述	值
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	<p>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。</p> <div style="display: flex; align-items: center;">  <p>重要</p> </div> <p>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。</p>	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	<p>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。</p> <div style="display: flex; align-items: center;">  <p>重要</p> </div> <p>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。</p>	Enabled 或 Disabled

参数	描述	值
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	<p>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。</p> <div style="display: flex; align-items: flex-start;"> <div style="width: 20px; height: 20px; background: repeating-linear-gradient(45deg, transparent, transparent 2px, black 2px, black 4px); margin-right: 10px;"></div> <div> <p>注意</p> <p>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 <i>Red Hat Operator 参考指南</i> 内容中的 <i>Cloud Credential Operator</i> 条目。</p> </div> </div>	Mint、Passthrough、Manual 或空字符串("")。
fips	<p>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。</p> <div style="display: flex; align-items: flex-start;"> <div style="width: 20px; height: 20px; background: repeating-linear-gradient(45deg, transparent, transparent 2px, black 2px, black 4px); margin-right: 10px; margin-bottom: 10px;"></div> <div> <p>重要</p> <p>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。</p> </div> </div> <div style="display: flex; align-items: flex-start;"> <div style="width: 20px; height: 20px; background: repeating-linear-gradient(45deg, transparent, transparent 2px, black 2px, black 4px); margin-right: 10px;"></div> <div> <p>注意</p> <p>如果使用 Azure File 存储，则无法启用 FIPS 模式。</p> </div> </div>	false 或 true

参数	描述	值
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。	<p>Internal 或 External。默认值为 External。</p> <p>在非云平台上不支持将此字段设置为 Internal。</p> <div style="display: flex; align-items: flex-start;"> <div style="width: 60px; height: 60px; background: repeating-linear-gradient(45deg, transparent, transparent 2px, black 2px, black 4px); margin-right: 10px;"></div> <div> <p>重要</p> <p>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。</p> </div> </div>
sshKey	<p>用于验证集群机器访问的 SSH 密钥或密钥。</p> <div style="display: flex; align-items: flex-start;"> <div style="width: 60px; height: 60px; background: repeating-linear-gradient(45deg, transparent, transparent 2px, black 2px, black 4px); margin-right: 10px;"></div> <div> <p>注意</p> <p>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。</p> </div> </div>	<p>一个或多个密钥。例如：</p> <pre>sshKey: <key1> <key2> <key3></pre>

1.4.14.4. 其他 Red Hat OpenStack Platform (RHOSP) 配置参数

下表描述了其他 RHOSP 配置参数：

表 1.23. 其他 RHOSP 参数

参数	描述	值
<code>compute.platform.openstack.rootVolume.size</code>	对于计算机器，以 GB 为单位表示的根卷大小。如果您不设置这个值，机器将使用临时存储。	整数，如 30 。
<code>compute.platform.openstack.rootVolume.type</code>	对于计算机器，根卷的类型。	字符串，如 performance 。
<code>controlPlane.platform.openstack.rootVolume.size</code>	对于 control plane 机器，以 GB 为单位表示的根卷大小。如果您不设置这个值，机器将使用临时存储。	整数，如 30 。
<code>controlPlane.platform.openstack.rootVolume.type</code>	对于 control plane 机器，根卷的类型。	字符串，如 performance 。
<code>platform.openstack.cloud</code>	要使用的 RHOSP 云的名称，来自于 <code>clouds.yaml</code> 文件中的云列表。	字符串，如 MyCloud 。
<code>platform.openstack.externalNetwork</code>	用于安装的 RHOSP 外部网络名称。	字符串，如 external 。
<code>platform.openstack.computeFlavor</code>	用于 control plane 和计算机器的 RHOSP 类别。	字符串，如 m1.xlarge 。

1.4.14.5. 可选 RHOSP 配置参数

下表描述了可选 RHOSP 配置参数：

表 1.24. 可选的 RHOSP 参数

参数	描述	值
<code>compute.platform.openstack.additionalNetworkIDs</code>	与计算机器关联的其他网络。不能为额外网络创建允许的地址对。	一个或多个 UUID 列表作为字符串。例如： fa806b2f-ac49-4bce-b9db-124bc64209bf 。
<code>compute.platform.openstack.additionalSecurityGroupIDs</code>	与计算机器关联的其他安全组。	一个或多个 UUID 列表作为字符串。例如： 7ee219f3-d2e9-48a1-96c2-e7429f1b0da7 。

参数	描述	值
compute.platform.openstack.zones	<p>RHOSP Compute (Nova) 可用区 (AZ) 在其中安装机器。如果没有设置此参数, 安装程序会依赖于配置了 RHOSP 管理员的 Nova 的默认设置。</p> <p>在使用 Kuryr 的集群上, RHOSP Octavia 不支持可用域。负载均衡器, 如果您使用 Amphora 供应商驱动程序, 则依赖 Amphora 虚拟机的 OpenShift Container Platform 服务不会根据此属性的值创建。</p>	字符串列表。例如: ["zone-1", "zone-2"]。
controlPlane.platform.openstack.additionalNetworkIDs	与 control plane 机器关联的额外网络。不能为额外网络创建允许的地址对。	一个或多个 UUID 列表作为字符串。例如: fa806b2f-ac49-4bce-b9db-124bc64209bf 。
controlPlane.platform.openstack.additionalSecurityGroupIDs	与 control plane 机器关联的其他安全组。	一个或多个 UUID 列表作为字符串。例如: 7ee219f3-d2e9-48a1-96c2-e7429f1b0da7 。
controlPlane.platform.openstack.zones	<p>RHOSP Compute (Nova) 可用区 (AZ) 在其中安装机器。如果没有设置此参数, 安装程序会依赖于配置了 RHOSP 管理员的 Nova 的默认设置。</p> <p>在使用 Kuryr 的集群上, RHOSP Octavia 不支持可用域。负载均衡器, 如果您使用 Amphora 供应商驱动程序, 则依赖 Amphora 虚拟机的 OpenShift Container Platform 服务不会根据此属性的值创建。</p>	字符串列表。例如: ["zone-1", "zone-2"]。
platform.openstack.clusterOSImage	<p>安装程序从中下载 RHCOS 镜像的位置。</p> <p>您必须设置此参数以便在受限网络中执行安装。</p>	<p>HTTP 或 HTTPS URL, 可选使用 SHA-256 checksum。</p> <p>例如: http://mirror.example.com/images/rhcos-43.81.201912131630.0-openstack.x86_64.qcow2.gz?sha256=ffebbd68e8a1f2a245ca19522c16c86f67f9ac8e4e0c1f0a812b068b16f7265d。该值也可以是现有 Glance 镜像的名称, 如 my-rhcos。</p>

参数	描述	值
platform.openstack.defaultMachinePlatform	默认机器池平台配置。	<pre>{ "type": "ml.large", "rootVolume": { "size": 30, "type": "performance" } }</pre>
platform.openstack.ingressFloatingIP	与 Ingress 端口关联的现有浮动 IP 地址。要使用此属性，还必须定义 platform.openstack.externalNetwork 属性。	IP 地址，如 128.0.0.1 。
platform.openstack.lbFloatingIP	与 API 负载均衡器关联的现有浮动 IP 地址。要使用此属性，还必须定义 platform.openstack.externalNetwork 属性。	IP 地址，如 128.0.0.1 。
platform.openstack.externalDNS	集群实例用于进行 DNS 解析的外部 DNS 服务器的 IP 地址。	一个 IP 地址列表作为字符串。例如， ["8.8.8.8", "192.168.1.12"] 。
platform.openstack.machinesSubnet	<p>集群节点使用的 RHOSP 子网的 UUID。在这个子网上创建节点和虚拟 IP (VIP) 端口。</p> <p>networking.machineNetwork 中的第一个项需要和 machinesSubnet 的值匹配。</p> <p>如果部署到自定义子网中，则无法将外部 DNS 服务器指定到 OpenShift Container Platform 安装程序。反之，把 DNS 添加到 RHOSP 的子网。</p>	作为字符串的 UUID。例如： fa806b2f-ac49-4bceb9db-124bc64209bf 。

1.4.14.6. RHOSP 部署中的自定义子网

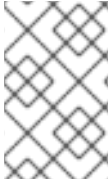
另外，您还可以在您选择的 Red Hat OpenStack Platform (RHOSP) 子网中部署集群。子网的 GUID 作为 **install-config.yaml** 文件中的 **platform.openstack.machinesSubnet** 的值传递。

此子网被用作集群的主子网，在其上创建节点和端口。

在使用自定义子网运行 OpenShift Container Platform 安装程序前，请验证：

- 目标网络和子网可用。

- 目标子网上启用了 DHCP。
- 您可提供在目标网络上有创建端口权限的安装程序凭证。
- 如果您的网络配置需要一个路由器，它会在 RHOSP 中创建。有些配置依赖于路由器来转换浮动 IP 地址。
- 您的网络配置不依赖于供应商网络。不支持提供商网络。

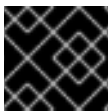


注意

默认情况下，API VIP 使用 x.x.x.5，Ingress VIP 从网络 CIDR 块获取 x.x.x.7。要覆盖这些默认值，为 DHCP 分配池以外的 **platform.openstack.apiVIP** 和 **platform.openstack.ingressVIP** 设置值。

1.4.14.7. 使用 Kuryr 的 RHOSP 的自定义 `install-config.yaml` 文件示例

要使用 Kuryr SDN 而不是默认的 OpenShift SDN 部署，您必须修改 `install-config.yaml` 文件，使其包含 **Kuryr** 作为所需的 **networking.networkType**，然后执行默认的 OpenShift Container Platform SDN 安装步骤。此示例 `install-config.yaml` 展示了所有可能的 Red Hat OpenStack Platform (RHOSP) 自定义选项。



重要

此示例文件仅供参考。您必须使用安装程序来获取 `install-config.yaml` 文件。

```
apiVersion: v1
baseDomain: example.com
clusterID: os-test
controlPlane:
  name: master
  platform: {}
  replicas: 3
compute:
- name: worker
  platform:
    openstack:
      type: ml.large
  replicas: 3
metadata:
  name: example
networking:
  clusterNetwork:
  - cidr: 10.128.0.0/14
    hostPrefix: 23
  machineNetwork:
  - cidr: 10.0.0.0/16
  serviceNetwork:
  - 172.30.0.0/16 1
  networkType: Kuryr
platform:
  openstack:
    cloud: mycloud
    externalNetwork: external
```

```

computeFlavor: m1.xlarge
lbFloatingIP: 128.0.0.1
trunkSupport: true ②
octaviaSupport: true ③
pullSecret: '{"auths": ...}'
sshKey: ssh-ed25519 AAAA...

```

- ① Amphora Octavia 驱动程序为每个负载均衡器创建两个端口。因此，安装程序创建的服务子网是由 **serviceNetwork** 属性值指定的 CIDR 的两倍。要防止 IP 地址冲突，则需要更大的范围。
- ② ③ 安装程序会自动发现 **trunkSupport** 和 **octaviaSupport**，因此无需设置它们。但是，如果您的环境不满足这两个要求，Kuryr SDN 将无法正常工作。需要使用中继来把 pod 连接到 RHOSP 网络，并且需要 Octavia 来创建 OpenShift Container Platform 服务。

1.4.14.8. Kuryr 端口池

Kuryr 端口池在待机时维护多个端口，用于创建 pod。

将端口保留在待机上可最大程度缩短 pod 创建时间。如果没有端口池，Kuryr 必须明确请求在创建或删除 pod 时创建或删除端口。

Kuryr 使用的 Neutron 端口是在绑定到命名空间的子网中创建的。这些 pod 端口也作为子端口添加到 OpenShift Container Platform 集群节点的主端口。

因为 Kuryr 将每个命名空间保留在单独的子网中，所以对于每个“命名空间-worker”对都会维护一个单独的端口池。

在安装集群前，您可以在 **cluster-network-03-config.yml** 清单文件中设置以下参数来配置端口池行为：

- **enablePortPoolsPrepopulation** 参数控制池预填充，它会强制 Kuryr 在创建时（如添加新主机或创建新命名空间时）将端口添加到池中。默认值为：**false**。
- **poolMinPorts** 参数是池中保留的最少可用端口的数量。默认值为：**1**。
- **poolMaxPorts** 参数是池中保留的最大可用端口数。如果值为 **0**，会禁用上限。这是默认的设置。
如果您的 OpenStack 端口配额较低，或者 pod 网络上的 IP 地址有限，请考虑设置此选项以确保删除不需要的端口。
- **poolBatchPorts** 参数定义一次可以创建的 Neutron 端口的最大数量。默认值为 **3**。

1.4.14.9. 在安装过程中调整 Kuryr 端口池

在安装过程中，您可以配置 Kuryr 如何管理 Red Hat OpenStack Platform (RHOSP) Neutron 端口，以控制 pod 创建的速度和效率。

先决条件

- 创建并修改 **install-config.yaml** 文件。

流程

1. 在命令行中创建清单文件：

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

- 1 对于 **<installation_directory>**，请指定含有集群的 **install-config.yaml** 文件的目录的名称。

2. 在 **<installation_directory>/manifests/** 目录下，创建一个名为 **cluster-network-03-config.yml** 的文件：

```
$ touch <installation_directory>/manifests/cluster-network-03-config.yml 1
```

- 1 对于 **<installation_directory>**，请指定包含集群的 **manifests/** 目录的目录名称。

创建该文件后，**manifests/** 目录中会包含多个网络配置文件，如下所示：

```
$ ls <installation_directory>/manifests/cluster-network-*
```

输出示例

```
cluster-network-01-crd.yml
cluster-network-02-config.yml
cluster-network-03-config.yml
```

3. 在编辑器中打开 **cluster-network-03-config.yml** 文件，并输入描述您想要的 Cluster Network Operator 配置的自定义资源(CR)：

```
$ oc edit networks.operator.openshift.io cluster
```

4. 编辑设置以满足您的要求。以下示例提供了以下文件：

```
apiVersion: operator.openshift.io/v1
kind: Network
metadata:
  name: cluster
spec:
  clusterNetwork:
  - cidr: 10.128.0.0/14
    hostPrefix: 23
  serviceNetwork:
  - 172.30.0.0/16
  defaultNetwork:
    type: Kuryr
    kuryrConfig:
      enablePortPoolsPrepopulation: false 1
      poolMinPorts: 1 2
      poolBatchPorts: 3 3
      poolMaxPorts: 5 4
      openstackServiceNetwork: 172.30.0.0/15 5
```

- 1 将 **enablePortPoolsPrepopulation** 的值设置为 **true** 以使 Kuryr 在创建命名空间或在集群中添加新节点后创建新 Neutron 端口。此设置引发 Neutron 端口配额，但可以缩短生成容器集所需的时间。默认值为：**false**。

- 2 如果池中的可用端口数量低于 **poolMinPorts** 的值，Kuryr 会为池创建新端口。默认值为 **1**。
- 3 **poolBatchPorts** 控制在可用端口数量低于 **poolMinPorts** 值时创建的新端口数量。默认值为 **3**。
- 4 如果池中的可用端口数量大于 **poolMaxPorts** 的值，Kuryr 会删除它们，直到数量与这个值匹配为止。将此值设置为 **0** 可禁用此上限，防止池缩小。默认值为 **0**。
- 5 **openStackServiceNetwork** 参数定义将 IP 地址分配到 RHOSP Octavia 的 LoadBalancer 的网络的 CIDR 范围。

如果此参数与 Amphora 驱动程序一起使用，则 Octavia 会为每个负载均衡器从这个网络获取两个 IP 地址：一个用于 OpenShift，另一个用于 VRRP 连接。由于这些 IP 地址分别由 OpenShift Container Platform 和 Neutron 管理，因此它们必须来自不同的池。因此，**openStackServiceNetwork** 的值必须至少是 **serviceNetwork** 值的两倍，**serviceNetwork** 的值必须与 **openStackServiceNetwork** 定义的范围完全重叠。

CNO 验证从此参数定义的范围获取的 VRRP IP 地址是否与 **serviceNetwork** 参数定义的范围不重叠。

如果没有设置此参数，CNO 将使用 **serviceNetwork** 的扩展值，它是前缀大小值减 1。

5. 保存 **cluster-network-03-config.yml** 文件，再退出文本编辑器。
6. 可选：备份 **manifests/cluster-network-03-config.yml** 文件。安装程序在创建集群时删除 **manifests/** 目录。

1.4.14.10. 为机器设置自定义子网

安装程序默认使用的 IP 范围可能与您在安装 OpenShift Container Platform 时创建的 Neutron 子网不匹配。如有必要，通过编辑安装配置文件来更新新机器的 CIDR 值。

先决条件

- 有 OpenShift Container Platform 安装程序生成的 **install-config.yaml** 文件。

流程

1. 在命令行中进入包含 **install-config.yaml** 的目录。
2. 在该目录中，运行脚本来编辑 **install-config.yaml** 文件或手动更新该文件：
 - 要使用脚本设置值，请运行：

```
$ python -c '
import yaml;
path = "install-config.yaml";
data = yaml.safe_load(open(path));
data["networking"]["machineNetwork"] = [{"cidr": "192.168.0.0/18"}]; 1
open(path, "w").write(yaml.dump(data, default_flow_style=False))'
```

- 1 插入一个与您指定的 Neutron 子网匹配的值，如 **192.0.2.0/24**。

- 要手动设置这个值，请打开该文件并将 **networking.machineCIDR** 的值设置为与您预期的 Neutron 子网匹配的内容。

1.4.14.11. 清空计算机器池

要进行使用您自己的基础架构的安装，请将安装配置文件中的计算机器数量设置为零。之后，您可以手动创建这些机器。

先决条件

- 有 OpenShift Container Platform 安装程序生成的 **install-config.yaml** 文件。

流程

1. 在命令行中进入包含 **install-config.yaml** 的目录。
2. 在该目录中，运行脚本来编辑 **install-config.yaml** 文件或手动更新该文件：
 - 要使用脚本设置值，请运行：

```
$ python -c '  
import yaml;  
path = "install-config.yaml";  
data = yaml.safe_load(open(path));  
data["compute"][0]["replicas"] = 0;  
open(path, "w").write(yaml.dump(data, default_flow_style=False))'
```

- 要手动设置值，打开文件并将 **compute.<first entry>.replicas** 的值设置为 **0**。

1.4.14.12. 修改网络类型

默认情况下，安装程序会选择 **OpenShiftSDN** 网络类型。要使用 Kuryr，请更改安装程序生成的安装配置文件中的值。

先决条件

- 有 OpenShift Container Platform 安装程序生成的 **install-config.yaml** 文件

流程

1. 在命令提示符中，进入包含 **install-config.yaml** 的目录。
2. 在该目录中，运行脚本来编辑 **install-config.yaml** 文件或手动更新该文件：
 - 要使用脚本设置值，请运行：

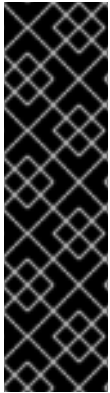
```
$ python -c '  
import yaml;  
path = "install-config.yaml";  
data = yaml.safe_load(open(path));  
data["networking"]["networkType"] = "Kuryr";  
open(path, "w").write(yaml.dump(data, default_flow_style=False))'
```

- 要手动设置这个值，打开该文件并将 **networking.networkType** 设置为 **"Kuryr"**。

1.4.15. 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. 删除定义 control plane 机器的 Kubernetes 清单文件以及计算机器集：

```
$ rm -f openshift/99_openshift-cluster-api_master-machines-*.yaml openshift/99_openshift-cluster-api_worker-machineset-*.yaml
```

由于您要自行创建和管理这些资源，因此不必初始化这些资源。

- 您可以使用机器 API 来保留机器集文件来创建计算机器，但您必须更新对其的引用，以匹配您的环境。
3. 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。
 4. 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

```

5. 将元数据文件的 `infraID` 键导出为环境变量：

```
$ export INFRA_ID=$(jq -r .infraID metadata.json)
```

提示

从 `metadata.json` 中提取 `infraID` 键，并将其用作您创建的所有 RHOSP 资源的前缀。通过这样做，您可以避免在同一项目进行多个部署时的名称冲突。

1.4.16. 准备 bootstrap Ignition 文件

OpenShift Container Platform 安装过程依赖于从 bootstrap Ignition 配置文件创建的 bootstrap 机器。

编辑该文件并上传该文件。然后，创建 Red Hat OpenStack Platform (RHOSP) 用来下载主文件的辅助 bootstrap Ignition 配置文件。

先决条件

- 您有安装程序生成的 bootstrap Ignition 文件，即 `bootstrap.ign`。
- 安装程序元数据文件中的基础架构 ID 被设置为环境变量 (`$INFRA_ID`)。
 - 如果未设置变量，请参阅 [创建 Kubernetes 清单和 Ignition 配置文件](#)。
- 可以使用 HTTP(S) 来存储 bootstrap ignition 文件。
 - 所记录的步骤使用 RHOSP 镜像服务 (Glance)，但也可以使用 RHOSP Storage 服务 (Swift)、Amazon S3、内部 HTTP 服务器或临时 Nova 服务器。

流程

1. 运行以下 Python 脚本。该脚本修改 bootstrap Ignition 文件，以设置主机名，并在运行时设置 CA 证书文件：

```
import base64
import json
import os
```



```

with open('bootstrap.ign', 'r') as f:
    ignition = json.load(f)

files = ignition['storage'].get('files', [])

infra_id = os.environ.get('INFRA_ID', 'openshift').encode()
hostname_b64 = base64.standard_b64encode(infra_id + b'-bootstrap\n').decode().strip()
files.append(
    {
        'path': '/etc/hostname',
        'mode': 420,
        'contents': {
            'source': 'data:text/plain;charset=utf-8;base64,' + hostname_b64
        }
    }
)

ca_cert_path = os.environ.get('OS_CACERT', "")
if ca_cert_path:
    with open(ca_cert_path, 'r') as f:
        ca_cert = f.read().encode()
        ca_cert_b64 = base64.standard_b64encode(ca_cert).decode().strip()

    files.append(
        {
            'path': '/opt/openshift/tls/cloud-ca-cert.pem',
            'mode': 420,
            'contents': {
                'source': 'data:text/plain;charset=utf-8;base64,' + ca_cert_b64
            }
        }
    )

ignition['storage']['files'] = files;

with open('bootstrap.ign', 'w') as f:
    json.dump(ignition, f)

```

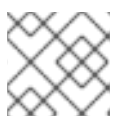
2. 使用 RHOSP CLI，创建使用 bootstrap Ignition 文件的镜像：

```
$ openstack image create --disk-format=raw --container-format=bare --file bootstrap.ign
<image_name>
```

3. 获取镜像的详情：

```
$ openstack image show <image_name>
```

请记录 **file** 值；它需要遵循 **v2/images/<image_ID>/file** 格式。



注意

验证您创建的镜像是否活跃。

4. 检索镜像服务的公共地址：

```
$ openstack catalog show image
```

- 将公共地址与镜像的 **file** 值合并，并在存储位置保存结果。位置遵循 **<image_service_public_URL>/v2/images/<image_ID>/file** 格式。
- 生成身份验证令牌并保存令牌 ID:

```
$ openstack token issue -c id -f value
```

- 将以下内容插入到名为 **\$INFRA_ID-bootstrap-ignition.json** 的文件中，并编辑位置拥有者以匹配您自己的值：

```
{
  "ignition": {
    "config": {
      "merge": [{
        "source": "<storage_url>", ❶
        "httpHeaders": [{
          "name": "X-Auth-Token", ❷
          "value": "<token_ID>" ❸
        }]
      }],
    },
    "security": {
      "tls": {
        "certificateAuthorities": [{
          "source": "data:text/plain;charset=utf-8;base64,<base64_encoded_certificate>" ❹
        }]
      }
    },
    "version": "3.1.0"
  }
}
```

- ❶ 将 **ignition.config.merge.source** 的值替换为 bootstrap Ignition 文件存储 URL。
- ❷ 在 **httpHeaders** 中将 **name** 设置为 **"X-Auth-Token"**。
- ❸ 在 **httpHeaders** 中将 **value** 设为您的令牌 ID。
- ❹ 如果 bootstrap Ignition 文件服务器使用自签名证书,请包括以 base64 编码的证书。

- 保存二级 Ignition 配置文件。

bootstrap Ignition 数据将在安装过程中传递给 RHOSP。



警告

bootstrap Ignition 文件包含敏感信息，如 **clouds.yaml** 凭证。确定您将其保存在安全的地方，并在完成安装后将其删除。

1.4.17. 在 RHOSP 上创建 control plane Ignition 配置文件

在您自己的基础架构的 Red Hat OpenStack Platform (RHOSP) 上安装 OpenShift Container Platform 需要 control plane Ignition 配置文件。您必须创建多个配置文件。



注意

与 bootstrap Ignition 配置一样，您必须明确为每个 control plane 机器定义主机名。

先决条件

- 来自安装程序元数据文件中的基础架构 ID 被设置为环境变量 (**\$INFRA_ID**)。
 - 如果未设置变量，请参阅“创建 Kubernetes 清单和 Ignition 配置文件”。

流程

- 在命令行中运行以下 Python 脚本：

```
$ for index in $(seq 0 2); do
  MASTER_HOSTNAME="$INFRA_ID-master-$index\n"
  python -c "import base64, json, sys;
  ignition = json.load(sys.stdin);
  storage = ignition.get('storage', {});
  files = storage.get('files', []);
  files.append({'path': '/etc/hostname', 'mode': 420, 'contents': {'source':
'data:text/plain;charset=utf-8;base64,' +
base64.standard_b64encode(b'$MASTER_HOSTNAME').decode().strip(), 'verification': {}},
'filesystem': 'root'});
  storage['files'] = files;
  ignition['storage'] = storage
  json.dump(ignition, sys.stdout) <master.ign >"$INFRA_ID-master-$index-ignition.json"
done
```

您现在有三个 control plane Ignition 文件：**<INFRA_ID>-master-0-ignition.json**、**<INFRA_ID>-master-1-ignition.json** 和 **<INFRA_ID>-master-2-ignition.json**。

1.4.18. 在 RHOSP 上创建网络资源

在您自己的基础架构的 Red Hat OpenStack Platform (RHOSP) 安装上创建 OpenShift Container Platform 所需的网络资源。为节省时间，可以运行提供的 Ansible playbook 来生成安全组、网络、子网、路由器和端口。

先决条件

- Python 3 已安装在您的机器上。
- 您下载了“下载 playbook 依赖项”中的模块。
- 下载了“下载安装 playbook”中的 playbook。

流程

1. 可选：为 **inventory.yaml** playbook 添加一个外部网络值：

inventory.yaml Ansible playbook 中的外部网络值示例

```
...
# The public network providing connectivity to the cluster. If not
# provided, the cluster external connectivity must be provided in another
# way.

# Required for os_api_fip, os_ingress_fip, os_bootstrap_fip.
os_external_network: 'external'
...
```



重要

如果没有为 **inventory.yaml** 文件中的 **os_external_network** 提供值，则必须确保虚拟机可以自行访问 Glance 和外部连接。

2. 可选：将外部网络和浮动 IP（FIP）地址值添加到 **inventory.yaml** playbook：

inventory.yaml Ansible playbook 中的 FIP 值示例

```
...
# OpenShift API floating IP address. If this value is non-empty, the
# corresponding floating IP will be attached to the Control Plane to
# serve the OpenShift API.
os_api_fip: '203.0.113.23'

# OpenShift Ingress floating IP address. If this value is non-empty, the
# corresponding floating IP will be attached to the worker nodes to serve
# the applications.
os_ingress_fip: '203.0.113.19'

# If this value is non-empty, the corresponding floating IP will be
# attached to the bootstrap machine. This is needed for collecting logs
# in case of install failure.
os_bootstrap_fip: '203.0.113.20'
```



重要

如果您没有为 **os_api_fip** 和 **os_ingress_fip** 定义值，则必须执行安装后的网络配置。

如果您没有为 **os_bootstrap_fip** 定义值，安装程序将无法从失败的安装中下载调试信息。

如需更多信息，请参阅“启用对环境的访问”。

3. 在命令行中，通过运行 **security-groups.yaml** playbook 来创建安全组：

```
$ ansible-playbook -i inventory.yaml security-groups.yaml
```

4. 在命令行中，通过运行 **network.yaml** playbook 来创建一个网络、子网和路由器：

```
$ ansible-playbook -i inventory.yaml network.yaml
```

- 5. 可选：如果要控制 Nova 服务器使用的默认解析程序，请运行 RHOSP CLI 命令：

```
$ openstack subnet set --dns-nameserver <server_1> --dns-nameserver <server_2>
"$INFRA_ID-nodes"
```

1.4.19. 在 RHOSP 上创建 bootstrap 机器

创建 bootstrap 机器，为其提供在 Red Hat OpenStack Platform (RHOSP) 上运行所需的网络访问权限。红帽提供了一个 Ansible playbook，您可运行它来简化此过程。

先决条件

- 您下载了"下载 playbook 依赖项"中的模块。
- 下载了"下载安装 playbook"中的 playbook。
- **inventory.yaml**、**common.yaml** 和 **bootstrap.yaml** Ansible playbook 位于一个通用目录中。
- 安装程序创建的 **metadata.json** 文件与 Ansible playbook 位于同一个目录中。

流程

1. 在命令行中，将工作目录改为 playbook 的位置。
2. 在命令行中运行 **bootstrap.yaml** playbook：

```
$ ansible-playbook -i inventory.yaml bootstrap.yaml
```

3. bootstrap 服务器可用后，查看日志以验证是否收到 Ignition 文件：

```
$ openstack console log show "$INFRA_ID-bootstrap"
```

1.4.20. 在 RHOSP 中创建 control plane 机器

使用您生成的 Ignition 配置文件创建三台 control plane 机器。红帽提供了一个 Ansible playbook，您可运行它来简化此过程。

先决条件

- 您下载了"下载 playbook 依赖项"中的模块。
- 下载了"下载安装 playbook"中的 playbook。
- 来自安装程序元数据文件中的基础架构 ID 被设置为环境变量 (**\$INFRA_ID**)。
- **inventory.yaml**、**common.yaml** 和 **control-plane.yaml** Ansible playbook 位于一个通用目录中。
- 您有三个在"Creating control plane Ignition 配置文件"中创建的 Ignition 文件。

流程

1. 在命令行中，将工作目录改为 playbook 的位置。

2. 如果 control plane Ignition 配置文件尚未位于工作目录中，将其复制到其中。
3. 在命令行中运行 **control-plane.yaml** playbook：

```
$ ansible-playbook -i inventory.yaml control-plane.yaml
```

4. 运行以下命令来监控 bootstrap 过程：

```
$ openshift-install wait-for bootstrap-complete
```

您会看到确认 control plane 机器正在运行并加入集群的消息：

```
INFO API v1.14.6+f9b5405 up
INFO Waiting up to 30m0s for bootstrapping to complete...
...
INFO It is now safe to remove the bootstrap resources
```

1.4.21. 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
```

1.4.22. 从 RHOSP 删除 bootstrap 资源

删除您不再需要的 bootstrap 资源。

先决条件

- 您下载了"下载 playbook 依赖项"中的模块。
- 下载了"下载安装 playbook"中的 playbook。
- **inventory.yaml**、**common.yaml** 和 **down-bootstrap.yaml** Ansible playbook 位于一个通用目录中。
- control plane 机器正在运行。
 - 如果您不知道机器的状态，请参阅"验证集群状态"。

流程

1. 在命令行中，将工作目录改为 playbook 的位置。
2. 在命令行中运行 **down-bootstrap.yaml** playbook：

```
$ ansible-playbook -i inventory.yaml down-bootstrap.yaml
```

bootstrap 端口、服务器和浮动 IP 地址会被删除。



警告

如果您之前没有禁用 bootstrap ignition 文件 URL，现在需要禁用。

1.4.23. 在 RHOSP 上创建计算机

启动 control plane 后，创建计算机。红帽提供了一个 Ansible playbook，您可运行它来简化此过程。

先决条件

- 您下载了"下载 playbook 依赖项"中的模块。
- 下载了"下载安装 playbook"中的 playbook。
- **inventory.yaml**、**common.yaml** 和 **compute-nodes.yaml** Ansible playbook 位于一个通用目录中。
- 安装程序创建的 **metadata.json** 文件与 Ansible playbook 位于同一个目录中。
- control plane 处于活跃状态。

流程

1. 在命令行中，将工作目录改为 playbook 的位置。
2. 在命令行中运行 playbook:

```
$ ansible-playbook -i inventory.yaml compute-nodes.yaml
```

后续步骤

- 批准机器的证书签名请求。

1.4.24. 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

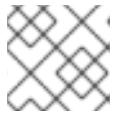
1. 确认集群可以识别这些机器：

```
$ 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：

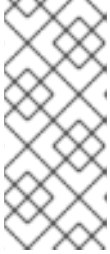
```
$ oc get csr
```

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-bootstrap** 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。

- 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
...
```

- 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：

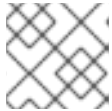
```
$ oc get csr -o go-template='{{range .items}}{{if not .status}}{{.metadata.name}}{\n"}\n{{end}}' | xargs oc adm certificate approve
```

6. 批准所有客户端和服务端 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](#)。

1.4.25. 验证安装是否成功

验证 OpenShift Container Platform 安装已完成。

先决条件

- 有安装程序 (**openshift-install**)

流程

- 在命令行中运行：

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

程序输出控制台 URL 以及管理员的登录信息。

1.4.26. 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](#)。

1.4.27. 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](#)。
- 如果您需要启用对节点端口的外部访问，[请使用节点端口配置集群流量](#)。
- 如果您没有将 RHOSP 配置为使用浮动 IP 地址接受应用程序流量，[使用浮动 IP 地址配置 RHOSP 访问](#)。

1.5. 在受限网络中的 OPENSTACK 上安装集群

在 OpenShift Container Platform 4.6 中，您可以通过创建安装发行内容的内部镜像在受限网络中的 Red Hat OpenStack Platform (RHOSP) 上安装集群。

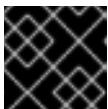


注意

在受限网络中安装仅支持安装程序置备的安装。

先决条件

- [在镜像主机上创建镜像 registry](#)，并获取您的 OpenShift Container Platform 版本的 `imageContentSources` 数据。



重要

由于安装介质位于堡垒主机上，因此请使用该计算机完成所有安装步骤。

- 查看有关 [OpenShift Container Platform 安装和更新流程](#) 的详细信息。
 - 通过咨询架构文档的[可用平台列表](#)来验证 OpenShift Container Platform 4.6 是否与您的 RHOSP 版本兼容。您还可以查看 [OpenShift Container Platform 在 RHOSP 中的支持](#) 来比较不同版本的平台支持。
- 验证您的网络配置不依赖于供应商网络。不支持提供商网络。
- 在 RHOSP 中启用了元数据服务。

1.5.1. 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。

1.5.1.1. 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。

1.5.2. 在 RHOSP 上安装 OpenShift Container Platform 的资源指南

您的 Red Hat OpenStack Platform (RHOSP) 配额需要满足以下条件才支持 OpenShift Container Platform 安装：

表 1.25. RHOSP 上默认 OpenShift Container Platform 集群的建议资源

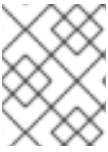
资源	值
浮动 IP 地址	3
端口	15
路由器	1
子网	1
RAM	112 GB
vCPUs	28
卷存储	275 GB
实例	7
安全组	3
安全组规则	60

集群或许能使用少于推荐数量的资源来运作，但其性能无法保证。



重要

如果 RHOSP 对象存储 (Swift) 可用, 并由具有 **swiftoperator** 角色的用户帐户执行, 它会作为 OpenShift Container Platform 镜像 registry 的默认后端。在这种情况下, 卷存储需要有 175GB。根据镜像 registry 的大小, Swift 空间要求会有所不同。



注意

默认情况下, 您的安全组和安全组规则配额可能较低。如果遇到问题, 请以 admin 的身份运行 **openstack quota set --secgroups 3 --secgroup-rules 60 <project>** 来提高配额。

OpenShift Container Platform 部署由 control plane 机器、计算机器和 bootstrap 机器组成。

1.5.2.1. control plane 机器

默认情况下, OpenShift Container Platform 安装过程会创建三台 control plane 机器。

每台机器都需要：

- 来自 RHOSP 配额的实例
- 来自 RHOSP 配额的端口
- 至少有 16 GB 内存、4 个 vCPU 和 100 GB 存储空间的类别

1.5.2.2. 计算机器

默认情况下, OpenShift Container Platform 安装过程会创建三台计算机器。

每台机器都需要：

- 来自 RHOSP 配额的实例
- 来自 RHOSP 配额的端口
- 至少有 8 GB 内存、2 个 vCPU 和 100 GB 存储空间的类别

提示

计算机器托管您在 OpenShift Container Platform 上运行的应用程序；运行数量应尽可能多。

1.5.2.3. bootstrap 机器

在安装时, 会临时置备 bootstrap 机器来支持 control plane。生产控制平面就绪后, bootstrap 机器会被取消置备。

bootstrap 机器需要：

- 来自 RHOSP 配额的实例
- 来自 RHOSP 配额的端口
- 至少有 16 GB 内存、4 个 vCPU 和 100 GB 存储空间的类别

1.5.3. 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。

1.5.4. 在 RHOSP 上启用 Swift

Swift 由具有 **swiftoperator** 角色的用户帐户操控。在运行安装程序前，将该角色添加到帐户。



重要

如果 [Red Hat OpenStack Platform \(RHOSP\) 对象存储服务](#) (通常称为 Swift) 可用，OpenShift Container Platform 会使用它作为镜像 registry 存储。如果无法使用，安装程序将依赖于 RHOSP 块存储服务，通常称为 Cinder。

如果 Swift 存在且您想要使用 Swift，则必须启用对其的访问。如果不存在，或者您不想使用它，请跳过这个部分。

先决条件

- 在目标环境中具有 RHOSP 管理员帐户
- 已安装 Swift 服务。
- 在 [Ceph RGW](#) 上启用了 **account in url** 选项。

流程

在 RHOSP 上启用 Swift：

1. 在 RHOSP CLI 中以管理员身份，将 **swiftoperator** 角色添加到要访问 Swift 的帐户：

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

您的 RHOSP 部署现可以使用 Swift 用于镜像 registry。

1.5.5. 为安装程序定义参数

OpenShift Container Platform 安装程序依赖于一个名为 **clouds.yaml** 的文件。该文件描述了 Red Hat OpenStack Platform (RHOSP) 配置参数，包括项目名称、登录信息和授权服务 URL。

流程

1. 创建 **clouds.yaml** 文件：

- 如果您的 RHOSP 发行版包含 Horizon web UI，请在该 UI 中生成 **clouds.yaml** 文件。

**重要**

请记住在 **auth** 字段中添加密码。您也可以把 secret 保存在 **clouds.yaml** 以外的一个独立的文件中。

- 如果您的 RHOSP 发行版不包含 Horizon Web UI，或者您不想使用 Horizon，请自行创建该文件。如需有关 **clouds.yaml** 的详细信息，请参阅 RHOSP 文档中的 [配置文件](#)。

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

2. 如果您的 RHOSP 安装使用自签名证书颁发机构 (CA) 证书进行端点身份验证：

- 将 CA 文件复制到您的机器中。
- 将机器添加到证书颁发机构信任捆绑包中：

```
$ sudo cp ca.crt.pem /etc/pki/ca-trust/source/anchors/
```

- 更新信任捆绑包：

```
$ sudo update-ca-trust extract
```

- 将 **cacerts** 键添加到 **clouds.yaml** 文件。该值必须是到 CA 证书的绝对路径，则其可以被非根用户访问：

```
clouds:
  shiftstack:
    ...
    cacert: "/etc/pki/ca-trust/source/anchors/ca.crt.pem"
```

提示

使用自定义 CA 证书运行安装程序后，您可以通过编辑 `cloud-provider-config` keymap 中的 `ca-cert.pem` 键的值来更新证书。在命令行中运行：

```
$ oc edit configmap -n openshift-config cloud-provider-config
```

3. 将 `clouds.yaml` 文件放在以下位置之一：
 - a. `OS_CLIENT_CONFIG_FILE` 环境变量的值
 - b. 当前目录
 - c. 特定于 Unix 的用户配置目录，如 `~/.config/openshift/clouds.yaml`
 - d. 特定于 Unix 的站点配置目录，如 `/etc/openshift/clouds.yaml`
安装程序会按照以上顺序搜索 `clouds.yaml`。

1.5.6. 为受限网络安装创建 RHCOS 镜像

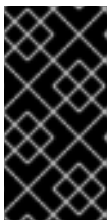
下载 Red Hat Enterprise Linux CoreOS (RHCOS) 镜像，以便在受限网络 Red Hat OpenStack Platform (RHOSP) 环境中安装 OpenShift Container Platform。

先决条件

- 获取 OpenShift Container Platform 安装程序。对于受限网络安装，该程序位于您的镜像 registry 主机上。

流程

1. 登录到红帽客户门户网站的[产品下载页](#)。
2. 在 **Version** 下，为 RHEL 8 选择 OpenShift Container Platform 4.6 的最新发行版本。



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。如果可用，请使用与 OpenShift Container Platform 版本匹配的镜像版本。

3. 下载 Red Hat Enterprise Linux CoreOS (RHCOS) - OpenStack Image (QCOW) 镜像。
4. 解压镜像。



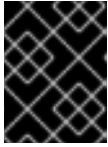
注意

您必须解压镜像，然后集群才能使用它。下载的文件名可能不包含压缩扩展名，如 `.gz` 或 `.tgz`。要找出是否以及如何压缩文件，请在命令行中输入：

```
$ file <name_of_downloaded_file>
```

5. 将您解压缩的镜像上传到堡垒服务器可访问的位置，如 Glance。例如：


```
$ openstack image create --file rhcos-44.81.202003110027-0-openstack.x86_64.qcow2 --
disk-format qcow2 rhcos-${RHCOS_VERSION}
```



重要

根据您的 RHOSP 环境，可能需要使用 **.raw** 或 **.qcow2** 格式下载镜像。如果使用 Ceph，则必须使用 **.raw** 格式。



警告

如果安装程序发现多个同名的镜像，它会随机选择其中之一。为避免这种行为，请在 RHOSP 中为资源创建唯一名称。

该镜像现在可用于受限安装。记录 OpenShift Container Platform 部署中使用的镜像名称或位置。

1.5.7. 创建安装配置文件

您可以自定义在 Red Hat OpenStack Platform (RHOSP) 上安装的 OpenShift Container Platform 集群。

先决条件

- 获取 OpenShift Container Platform 安装程序以及集群的 pull secret。对于受限网络安装，这些文件位于您的堡垒主机上。
- 具有创建镜像容器镜像仓库 (registry) 时生成的 **imageContentSources** 值。
- 获取您的镜像 registry 的证书内容。
- 检索 Red Hat Enterprise Linux CoreOS (RHCOS) 镜像，并将其上传到可访问的位置。

流程

1. 创建 **install-config.yaml** 文件。
 - a. 更改到包含安装程序的目录，再运行以下命令：

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

- 1** 对于 **<installation_directory>**，请指定用于保存安装程序所创建的文件目录名称。



重要

指定一个空目录。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。

- b. 在提示符处，提供您的云的配置详情：
 - i. 可选：选择用来访问集群机器的 SSH 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。

- ii. 选择 **openstack** 作为目标平台。
 - iii. 指定用于安装集群的 Red Hat OpenStack Platform (RHOSP) 外部网络名称。
 - iv. 指定用于从外部访问 OpenShift API 的浮动 IP 地址。
 - v. 指定至少有 16 GB RAM 用于 control plane 和计算节点的 RHOSP 类别。
 - vi. 选择集群要部署到的基域。所有 DNS 记录都将是这个基域的子域，并包含集群名称。
 - vii. 为集群输入一个名称。名称不能多于 14 个字符。
 - viii. 粘贴 [Red Hat OpenShift Cluster Manager](#) 中的 **pull secret**。
2. 在 **install-config.yaml** 文件中，将 **platform.openstack.clusterOSImage** 的值设置为镜像位置或名称。例如：

```
platform:
  openstack:
    clusterOSImage: http://mirror.example.com/images/rhcos-43.81.201912131630.0-
    openstack.x86_64.qcow2.gz?
    sha256=ffebbd68e8a1f2a245ca19522c16c86f67f9ac8e4e0c1f0a812b068b16f7265d
```

3. 编辑 **install-config.yaml** 文件，以提供在受限网络中安装所需的其他信息。

- a. 更新 **pullSecret** 值，使其包含 registry 的身份验证信息：

```
pullSecret: '{"auths":{"<mirror_host_name>:5000": {"auth": "<credentials>","email":
  "you@example.com"}}}'
```

对于 **<mirror_host_name>**，请指定您在镜像 registry 证书中指定的 registry 域名；对于 **<credentials>**，请指定您的镜像 registry 的 base64 编码用户名和密码。

- b. 添加 **additionalTrustBundle** 参数和值。

```
additionalTrustBundle: |
  -----BEGIN CERTIFICATE-----
  /-----END CERTIFICATE-----
```

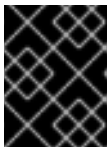
该值必须是您用于镜像 registry 的证书文件内容，可以是现有的可信证书颁发机构或您为镜像 registry 生成的自签名证书。

- c. 添加镜像内容资源，如下例所示：

```
imageContentSources:
- mirrors:
  - <mirror_host_name>:5000/<repo_name>/release
    source: quay.example.com/openshift-release-dev/ocp-release
- mirrors:
  - <mirror_host_name>:5000/<repo_name>/release
    source: registry.example.com/ocp/release
```

要完成这些值，请使用您在创建镜像容器镜像仓库（registry）时记录的 **imageContentSources**。

- 对需要的 **install-config.yaml** 文件做任何其他修改。您可以在 **安装配置参数** 部分中找到有关可用参数的更多信息。
- 备份 **install-config.yaml** 文件，以便用于安装多个集群。



重要

install-config.yaml 文件会在安装过程中消耗掉。如果要重复使用此文件，必须现在备份。

1.5.7.1. 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

- 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。

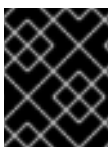
1.5.7.2. 安装配置参数

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。

1.5.7.2.1. 所需的配置参数

下表描述了所需的安装配置参数：

表 1.26. 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 。该字符串长度必须为 14 个字符或更少。
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>

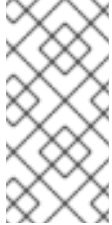
1.5.7.2.2. 网络配置参数

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 1.27. 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： <pre>networking: clusterNetwork: - cidr: 10.128.0.0/14 hostPrefix: 23</pre>
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： <pre>networking: serviceNetwork: - 172.30.0.0/16</pre>
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。	一个对象数组。例如： <pre>networking: machineNetwork: - cidr: 10.0.0.0/16</pre>

参数	描述	值
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 。	<p>CIDR 表示法中的 IP 网络块。</p> <p>例如：10.0.0.0/16。</p> <div style="display: flex; align-items: center;">  <div> <p>注意</p> <p>将 networking.machineNetwork 设置为与首选 NIC 所在的 CIDR 匹配。</p> </div> </div>



1.5.7.2.3. 可选配置参数

下表描述了可选安装配置参数：

表 1.28. 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	<p>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。</p> <div style="display: flex; align-items: center;">  <div> <p>重要</p> <p>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。</p> </div> </div>	Enabled 或 Disabled
compute.name	使用 compute 时需要此值。机器池的名称。	worker

参数	描述	值
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、o virt、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	<p>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。</p> <div style="display: flex; align-items: center;">  <div> <p>重要</p> <p>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。</p> </div> </div>	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、o virt、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。

参数	描述	值
credentialsMode	<p>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。</p>  <p>注意</p> <p>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 <i>Red Hat Operator 参考指南</i> 内容中的 <i>Cloud Credential Operator</i> 条目。</p>	Mint、Passthrough、Manual 或空字符串(“”)。
fips	<p>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。</p>  <p>重要</p> <p>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。</p>  <p>注意</p> <p>如果使用 Azure File 存储，则无法启用 FIPS 模式。</p>	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

参数	描述	值
imageContentSource.s.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。	<p>Internal 或 External。默认值为 External。</p> <p>在非云平台上不支持将此字段设置为 Internal。</p> <div style="display: flex; align-items: flex-start;">  <div> <p>重要</p> <p>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。</p> </div> </div>
sshKey	<p>用于验证集群机器访问的 SSH 密钥或密钥。</p> <div style="display: flex; align-items: flex-start;">  <div> <p>注意</p> <p>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。</p> </div> </div>	<p>一个或多个密钥。例如：</p> <pre>sshKey: <key1> <key2> <key3></pre>

1.5.7.2.4. 其他 Red Hat OpenStack Platform (RHOSP) 配置参数

下表描述了其他 RHOSP 配置参数：

表 1.29. 其他 RHOSP 参数

参数	描述	值
compute.platform.openstack.rootVolume.size	对于计算机器，以 GB 为单位的根卷大小。如果您不设置这个值，机器将使用临时存储。	整数，如 30 。
compute.platform.openstack.rootVolume.type	对于计算机器，根卷的类型。	字符串，如 performance 。

参数	描述	值
<code>controlPlane.platform.openstack.rootVolume.size</code>	对于 control plane 机器，以 GB 为单位表示的根卷大小。如果您不设置这个值，机器将使用临时存储。	整数，如 30 。
<code>controlPlane.platform.openstack.rootVolume.type</code>	对于 control plane 机器，根卷的类型。	字符串，如 performance 。
<code>platform.openstack.cloud</code>	要使用的 RHOSP 云的名称，来自于 <code>clouds.yaml</code> 文件中的云列表。	字符串，如 MyCloud 。
<code>platform.openstack.externalNetwork</code>	用于安装的 RHOSP 外部网络名称。	字符串，如 external 。
<code>platform.openstack.computeFlavor</code>	用于 control plane 和计算机器的 RHOSP 类别。	字符串，如 m1.xlarge 。

1.5.7.2.5. 可选 RHOSP 配置参数

下表描述了可选 RHOSP 配置参数：

表 1.30. 可选的 RHOSP 参数

参数	描述	值
<code>compute.platform.openstack.additionalNetworkIDs</code>	与计算机器关联的其他网络。不能为额外网络创建允许的地址对。	一个或多个 UUID 列表作为字符串。例如： fa806b2f-ac49-4bce-b9db-124bc64209bf 。
<code>compute.platform.openstack.additionalSecurityGroupIDs</code>	与计算机器关联的其他安全组。	一个或多个 UUID 列表作为字符串。例如： 7ee219f3-d2e9-48a1-96c2-e7429f1b0da7 。

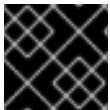
参数	描述	值
compute.platform.openstack.zones	<p>RHOSP Compute (Nova) 可用区 (AZ) 在其中安装机器。如果没有设置此参数, 安装程序会依赖于配置了 RHOSP 管理员的 Nova 的默认设置。</p> <p>在使用 Kuryr 的集群上, RHOSP Octavia 不支持可用域。负载均衡器, 如果您使用 Amphora 供应商驱动程序, 则依赖 Amphora 虚拟机的 OpenShift Container Platform 服务不会根据此属性的值创建。</p>	字符串列表。例如: ["zone-1", "zone-2"]。
controlPlane.platform.openstack.additionalNetworkIDs	与 control plane 机器关联的额外网络。不能为额外网络创建允许的地址对。	一个或多个 UUID 列表作为字符串。例如: fa806b2f-ac49-4bce-b9db-124bc64209bf 。
controlPlane.platform.openstack.additionalSecurityGroupIDs	与 control plane 机器关联的其他安全组。	一个或多个 UUID 列表作为字符串。例如: 7ee219f3-d2e9-48a1-96c2-e7429f1b0da7 。
controlPlane.platform.openstack.zones	<p>RHOSP Compute (Nova) 可用区 (AZ) 在其中安装机器。如果没有设置此参数, 安装程序会依赖于配置了 RHOSP 管理员的 Nova 的默认设置。</p> <p>在使用 Kuryr 的集群上, RHOSP Octavia 不支持可用域。负载均衡器, 如果您使用 Amphora 供应商驱动程序, 则依赖 Amphora 虚拟机的 OpenShift Container Platform 服务不会根据此属性的值创建。</p>	字符串列表。例如: ["zone-1", "zone-2"]。

参数	描述	值
platform.openstack.clusterOSImage	<p>安装程序从中下载 RHCOS 镜像的位置。</p> <p>您必须设置此参数以便在受限网络中执行安装。</p>	<p>HTTP 或 HTTPS URL，可选使用 SHA-256 checksum。</p> <p>例如： http://mirror.example.com/images/rhcos-43.81.201912131630.0-openstack.x86_64.qcow2.gz?sha256=ffebbd68e8a1f2a245ca19522c16c86f67f9ac8e4e0c1f0a812b068b16f7265d。该值也可以是现有 Glance 镜像的名称，如 my-rhcos。</p>
platform.openstack.defaultMachinePlatform	默认机器池平台配置。	<pre>{ "type": "ml.large", "rootVolume": { "size": 30, "type": "performance" } }</pre>
platform.openstack.ingressFloatingIP	<p>与 Ingress 端口关联的现有浮动 IP 地址。要使用此属性，还必须定义 platform.openstack.externalNetwork 属性。</p>	IP 地址，如 128.0.0.1 。
platform.openstack.lbFloatingIP	<p>与 API 负载均衡器关联的现有浮动 IP 地址。要使用此属性，还必须定义 platform.openstack.externalNetwork 属性。</p>	IP 地址，如 128.0.0.1 。
platform.openstack.externalDNS	集群实例用于进行 DNS 解析的外部 DNS 服务器的 IP 地址。	一个 IP 地址列表作为字符串。例如， ["8.8.8.8", "192.168.1.12"] 。

参数	描述	值
platform.openstack.machinesSubnet	<p>集群节点使用的 RHOSP 子网的 UUID。在这个子网上创建节点和虚拟 IP (VIP) 端口。</p> <p>networking.machineNetwork 中的第一个项需要和 machinesSubnet 的值匹配。</p> <p>如果部署到自定义子网中，则无法将外部 DNS 服务器指定到 OpenShift Container Platform 安装程序。反之，把 DNS 添加到 RHOSP 的子网。</p>	<p>作为字符串的 UUID。例如：fa806b2f-ac49-4bceb9db-124bc64209bf。</p>

1.5.7.3. 受限 OpenStack 安装的自定义 install-config.yaml 文件示例

此示例 **install-config.yaml** 展示了所有可能的 Red Hat OpenStack Platform (RHOSP) 自定义选项。



重要

此示例文件仅供参考。您必须使用安装程序来获取 **install-config.yaml** 文件。

```

apiVersion: v1
baseDomain: example.com
clusterID: os-test
controlPlane:
  name: master
  platform: {}
  replicas: 3
compute:
- name: worker
  platform:
    openstack:
      type: ml.large
    replicas: 3
metadata:
  name: example
networking:
  clusterNetwork:
  - cidr: 10.128.0.0/14
    hostPrefix: 23
  machineCIDR: 10.0.0.0/16
  serviceNetwork:
  - 172.30.0.0/16
  networkType: OpenShiftSDN
platform:
  openstack:
    region: region1
    cloud: mycloud
    externalNetwork: external

```



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

其中：

installation_directory

指定包含集群的 `install-config.yaml` 文件的目录名称。

3. 打开 `manifests/99_openshift-cluster-api_worker-machineset-0.yaml`，这是 **MachineSet** 定义文件。
4. 将属性 `serverGroupID` 添加到 `spec.template.spec.providerSpec.value` 属性下的定义中。例如：

```
apiVersion: machine.openshift.io/v1beta1
kind: MachineSet
metadata:
  labels:
    machine.openshift.io/cluster-api-cluster: <infrastructure_ID>
    machine.openshift.io/cluster-api-machine-role: <node_role>
    machine.openshift.io/cluster-api-machine-type: <node_role>
  name: <infrastructure_ID>-<node_role>
  namespace: openshift-machine-api
spec:
  replicas: <number_of_replicas>
  selector:
    matchLabels:
      machine.openshift.io/cluster-api-cluster: <infrastructure_ID>
      machine.openshift.io/cluster-api-machineset: <infrastructure_ID>-<node_role>
  template:
    metadata:
      labels:
        machine.openshift.io/cluster-api-cluster: <infrastructure_ID>
        machine.openshift.io/cluster-api-machine-role: <node_role>
        machine.openshift.io/cluster-api-machine-type: <node_role>
        machine.openshift.io/cluster-api-machineset: <infrastructure_ID>-<node_role>
    spec:
      providerSpec:
        value:
          apiVersion: openstackproviderconfig.openshift.io/v1alpha1
          cloudName: openstack
          cloudsSecret:
            name: openstack-cloud-credentials
            namespace: openshift-machine-api
          flavor: <nova_flavor>
          image: <glance_image_name_or_location>
          serverGroupID: aaaaaaaa-bbbb-cccc-dddd-eeeeeeeeeeeeee 1
          kind: OpenstackProviderSpec
          networks:
            - filter: {}
            subnets:
              - filter:
                  name: <subnet_name>
                  tags: openshiftClusterID=<infrastructure_ID>
          securityGroups:
            - filter: {}
```



```

name: <infrastructure_ID>-<node_role>
serverMetadata:
  Name: <infrastructure_ID>-<node_role>
  openshiftClusterID: <infrastructure_ID>
tags:
- openshiftClusterID=<infrastructure_ID>
trunk: true
userDataSecret:
  name: <node_role>-user-data
availabilityZone: <optional_openstack_availability_zone>

```

1 在此处添加服务器组的 UUID。

5. 可选：备份 `manifests/99_openshift-cluster-api_worker-machineset-0.yaml` 文件。创建集群时，安装程序会删除 `manifests/` 目录。

安装集群时，安装程序将使用您修改的 **MachineSet** 定义在 RHOSP 服务器组中创建计算机。

1.5.9. 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。

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。



注意

您必须使用一个本地密钥，而不要使用在特定平台上配置的密钥，如 [AWS 密钥对](#)。

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：

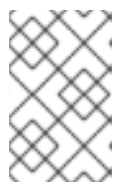
```

$ 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

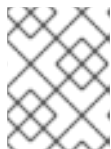
如果您计划在 **x86_64** 架构中安装使用 FIPS 验证的/Modules in Process 加密库的 OpenShift Container Platform 集群，不要创建使用 **ed25519** 算法的密钥。反之，创建一个使用 **rsa** 或 **ecdsa** 算法的密钥。

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。

1.5.10. 启用对环境的访问

在部署时，所有 OpenShift Container Platform 机器都是在 Red Hat OpenStack Platform (RHOSP) 租户网络中创建的。因此，大多数 RHOSP 部署中都无法直接访问它们。

您可以在安装过程中使用浮动 IP 地址（FIP）来配置 OpenShift Container Platform API 和应用程序访问。您也可以在没有配置 FIP 的情况下完成安装，但安装程序不会配置一种从外部访问 API 或应用程序的方法。

1.5.10.1. 启用通过浮动 IP 地址进行访问

创建浮动 IP（FIP）地址，用于从外部访问 OpenShift Container Platform API 和集群应用程序。

流程

1. 使用 Red Hat OpenStack Platform (RHOSP) CLI，创建 API FIP：

```
$ openstack floating ip create --description "API <cluster_name>.<base_domain>"
<external_network>
```

2. 使用 Red Hat OpenStack Platform (RHOSP) CLI，创建应用程序或 Ingress，FIP：

```
$ openstack floating ip create --description "Ingress <cluster_name>.<base_domain>"
<external_network>
```

- 向用于 API 和 Ingress FIP 的 DNS 服务器添加符合这些模式的记录：

```
api.<cluster_name>.<base_domain>. IN A <API_FIP>
*.apps.<cluster_name>.<base_domain>. IN A <apps_FIP>
```

注意

如果您不控制 DNS 服务器，您可以通过将集群域名（如以下内容）添加到 `/etc/hosts` 文件中来访问集群：

- `<api_floating_ip> api.<cluster_name>.<base_domain>`
- `<application_floating_ip> grafana-openshift-monitoring.apps.<cluster_name>.<base_domain>`
- `<application_floating_ip> prometheus-k8s-openshift-monitoring.apps.<cluster_name>.<base_domain>`
- `<application_floating_ip> oauth-openshift.apps.<cluster_name>.<base_domain>`
- `<application_floating_ip> console-openshift-console.apps.<cluster_name>.<base_domain>`
- `application_floating_ip integrate-oauth-server-openshift-authentication.apps.<cluster_name>.<base_domain>`

`/etc/hosts` 文件中的集群域名授予对本地集群的 Web 控制台和监控界面的访问权限。您还可以使用 `kubectl` 或 `oc`。您可以使用指向 `<application_floating_ip>` 的额外条目来访问用户应用程序。此操作使 API 和应用程序可供您访问，不适用于生产部署，但允许对开发和测试进行安装。

- 将 FIP 添加到 `install-config.yaml` 文件，将其作为以下参数的值：

- `platform.openstack.ingressFloatingIP`
- `platform.openstack.lbFloatingIP`

如果使用这些值，还必须在 `install-config.yaml` 文件中输入一个外部网络作为 `platform.openstack.externalNetwork` 参数的值。

提示

您可以通过分配浮动 IP 地址并更新防火墙配置，使 OpenShift Container Platform 资源在集群之外可用。

1.5.10.2. 完成没有浮动 IP 地址的安装

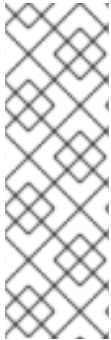
您可以在不提供浮动 IP 地址的情况下在 Red Hat OpenStack Platform (RHOSP) 上安装 OpenShift Container Platform。

在 `install-config.yaml` 文件中，不要定义以下参数：

- `platform.openstack.ingressFloatingIP`
- `platform.openstack.lbFloatingIP`

如果您无法提供外部网络，也可以将 `platform.openstack.externalNetwork` 留空。如果没有为 `platform.openstack.externalNetwork` 提供值，则不会为您创建路由器。如果没有额外的操作，安装程序将无法从 Glance 检索镜像。您必须自行配置外部连接。

如果在因为缺少浮动 IP 地址或名称解析而无法访问集群 API 的系统中运行安装程序时，安装会失败。要防止安装失败，可以使用代理网络或者从与您的机器位于同一网络的系统中运行安装程序。



注意

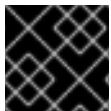
您可以通过为 API 和 Ingress 端口创建 DNS 记录来启用名称解析。例如：

```
api.<cluster_name>.<base_domain>. IN A <api_port_IP>
*.apps.<cluster_name>.<base_domain>. IN A <ingress_port_IP>
```

如果您不控制 DNS 服务器，可以改为将记录添加到 `/etc/hosts` 文件中。此操作使 API 可供您自己访问，不适用于生产部署。这可用于进行开发和测试的安装。

1.5.11. 部署集群

您可以在兼容云平台中安装 OpenShift Container Platform。



重要

安装程序的 `create cluster` 命令只能在初始安装过程中运行一次。

先决条件

- 获取 OpenShift Container Platform 安装程序以及集群的 pull secret。

流程

1. 更改为包含安装程序的目录并初始化集群部署：

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

1 对于 `<installation_directory>`，请指定自定义 `./install-config.yaml` 文件的位置。

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



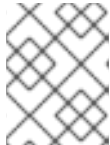
注意

如果您在主机上配置的云供应商帐户没有足够的权限来部署集群，安装过程将会停止，并且显示缺少的权限。

集群部署完成后，终端会显示访问集群的信息，包括指向其 Web 控制台的链接和 `kubeadmin` 用户的凭证。

输出示例

```
...
INFO Install complete!
INFO To access the cluster as the system:admin user when using 'oc', run 'export
KUBECONFIG=/home/myuser/install_dir/auth/kubeconfig'
INFO Access the OpenShift web-console here: https://console-openshift-
console.apps.mycluster.example.com
INFO Login to the console with user: "kubeadmin", and password: "4vYBz-Ee6gm-ymBZj-
Wt5AL"
INFO Time elapsed: 36m22s
```



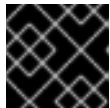
注意

当安装成功时，集群访问和凭证信息还会输出到 `<installation_directory>/openshift_install.log`。



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。



重要

您不得删除安装程序或安装程序所创建的文件。需要这两者才能删除集群。

1.5.12. 验证集群状态

您可以在安装过程中或安装后验证 OpenShift Container Platform 集群的状态：

流程

1. 在集群环境中，导出管理员的 kubeconfig 文件：

```
$ 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。

`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。

2. 查看部署后创建的 control plane 和计算机器：

```
$ oc get nodes
```

3. 查看集群的版本：

-

```
$ oc get clusterversion
```

4. 查看 Operator 的状态：

```
$ oc get clusteroperator
```

5. 查看集群中的所有正在运行的 pod:

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

1.5.13. 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
```

其他资源

- 如需有关访问和了解 OpenShift Container Platform Web 控制台的更多信息，请参阅[访问 Web 控制台](#)。

1.5.14. 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。

1.5.15. 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](#)。

1.5.16. 后续步骤

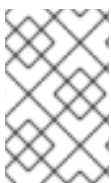
- [自定义集群](#)。
- 如果您用来安装集群的镜像 registry 具有一个可信任的 CA，通过[配置额外的信任存储](#)将其添加到集群中。
- 如果需要，您可以[选择不使用远程健康报告](#)。
- 为 Cluster Samples Operator 和 **must-gather** 工具[配置镜像流](#)。
- 了解如何在[受限网络中使用 Operator Lifecycle Manager \(OLM\)](#)。
- 如果您没有将 RHOSP 配置为使用浮动 IP 地址接受应用程序流量，[使用浮动 IP 地址配置 RHOSP 访问](#)。

1.6. 在 OPENSTACK 上卸载集群

您可以删除部署到 Red Hat OpenStack Platform (RHOSP) 的集群。

1.6.1. 删除使用安装程序置备的基础架构的集群

您可以从云中删除使用安装程序置备的基础架构的集群。



注意

卸载后，检查云供应商是否有没有被正确移除的资源，特别是 User Provisioned Infrastructure (UPI) 集群。可能存在安装程序没有创建的资源，或者安装程序无法访问的资源。

先决条件

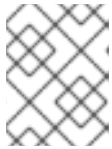
- 有部署集群时所用的安装程序副本。
- 有创建集群时安装程序所生成的文件。

流程

1. 在用来安装集群的计算机中包含安装程序的目录中，运行以下命令：

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

- 1** 对于 **<installation_directory>**，请指定安装文件保存到的目录的路径。
- 2** 要查看不同的详情，请指定 **warn**、**debug** 或 **error**，而不要指定 **info**。



注意

您必须为集群指定包含集群定义文件的目录。安装程序需要此目录中的 **metadata.json** 文件来删除集群。

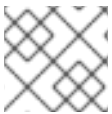
2. 可选：删除 **<installation_directory>** 目录和 OpenShift Container Platform 安装程序。

1.7. 从您自己的基础架构中卸载 RHOSP 上的集群

您可以删除在用户置备的基础架构上部署到 Red Hat OpenStack Platform (RHOSP) 中的集群。

1.7.1. 下载 **playbook** 的依赖项

用于简化从用户置备的基础架构中移除过程的 Ansible **playbook** 需要几个 Python 模块。在您要进行这个操作的机器上添加模块的仓库，然后下载它们。



注意

这些说明假设您使用 Red Hat Enterprise Linux (RHEL) 8。

先决条件

- Python 3 已安装在您的机器上。

流程

1. 在命令行中添加软件仓库：
 - a. 使用 Red Hat Subscription Manager 注册：

```
$ sudo subscription-manager register # If not done already
```

- b. 获取最新的订阅数据：

```
$ sudo subscription-manager attach --pool=$YOUR_POOLID # If not done already
```


c. 禁用当前的软件仓库：

```
$ sudo subscription-manager repos --disable=* # If not done already
```

d. 添加所需的软件仓库：

```
$ sudo subscription-manager repos \
  --enable=rhel-8-for-x86_64-baseos-rpms \
  --enable=openstack-16-tools-for-rhel-8-x86_64-rpms \
  --enable=ansible-2.9-for-rhel-8-x86_64-rpms \
  --enable=rhel-8-for-x86_64-appstream-rpms
```

2. 安装模块：

```
$ sudo yum install python3-openstackclient ansible python3-openstacksdk
```

3. 确保 **python** 命令指向 **python3**:

```
$ sudo alternatives --set python /usr/bin/python3
```

1.7.2. 从使用您自己的基础架构的 RHOSP 中删除集群

您可以在使用您自己的基础架构的 Red Hat OpenStack Platform (RHOSP) 上删除 OpenShift Container Platform 集群。要快速完成移除过程，请运行多个 Ansible playbook。

先决条件

- Python 3 已安装在您的机器上。
- 您下载了"下载 playbook 依赖项"中的模块。
- 您有用于安装集群的 playbook。
- 已修改前缀为 **down-** 的 playbook，以反映您对相应安装 playbook 所做的任何更改。例如，对 **bootstrap.yaml** 文件的改变会反映在 **down-bootstrap.yaml** 文件中。
- 所有 playbook 都位于一个通用目录中。

流程

1. 在命令行中运行您下载的 playbook：

```
$ ansible-playbook -i inventory.yaml \
  down-bootstrap.yaml \
  down-control-plane.yaml \
  down-compute-nodes.yaml \
  down-load-balancers.yaml \
  down-network.yaml \
  down-security-groups.yaml
```

2. 删除您为 OpenShift Container Platform 安装所做的任何 DNS 记录更改。

OpenShift Container Platform 被从您的基础架构中删除。

